To the Graduate Council:

I am submitting herewith a thesis written by Janet H. Carney entitled "PREP: A Programmable Remote Plotting Controller." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

J. Michael Moshell, Major Professor

We have read this thesis
and recommend its acceptance:

Martha R O'Kennon

Ronald P Leirnins

Accepted for the Council:

L Evan Peck

Vice Chancellor
Graduate Studies and Research

PREP: A PROGRAMMABLE REMOTE PLOTTING CONTROLLER

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Janet H. Carney

December 1981

3055450

ACKNOWLEDGMENTS

The device described in this thesis was developed for the Computer Sciences Organization of the Oak Ridge National Laboratory, which is operated by Union Carbide Corporation for the U.S. Department of Energy. The author wishes to thank Kathy E. Wyble, ORNL Text Management supervisor, and Richard D. McCulloch, ORNL Information Systems manager, whose support made this work possible.

Thanks are due also for the suggestions and comments made by Ronald P. Leinius and the computer programs written by Homer K. Meier. Fern E. Stooksbury is appreciated for typing the rough and final manuscript.

Special thanks are given to Robert G. LeBlanc for the many hours spent sharing his expertise in microcomputer interfacing and carefully reviewing thesis drafts.

ABSTRACT

This paper describes the design and implementation of a controller used to remotely interface a DEC PDP 11/70 computer to a VERSATEC electrostatic printer/plotter. The programmable remote plotting (PREP) controller consists of two microcomputers. One computer contains a Z80A microprocessor with programmable memory and serial/parallel I/O interfaces. The other computer contains the sophisticated 16-bit Z8000 microprocessor with extensive memory addressing. Two of the controller's main tasks are (1) transmit textual information or plot data from the host computer to the plotter; (2) download and execute programs from the host. Detailed hardware configuration and software techniques used to accomplish these tasks are supplied.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
	ORNL Word Processing	1
	Possible Enhancements	5
	The PREP Controller	6
II.	COMPONENT DESCRIPTIONS	9
	Host Computer	9
	Master Microcomputer	9
	Slave Microcomputer	12
	Plotter	13
III.	SYSTEM CONFIGURATION	16
	Serial I/O Port	16
	Parallel I/O Ports	23
	Interrupt Structure	28
	MULTIBUS	30
	Other Connections	32
IV.	REMOTE DATA COMMUNICATIONS	33
	Receiving Data	33
	Transmitting Data	38
	I/O Buffer Structure	42
V.	THE PREP PROCESSOR	47
	Plotter Commands	47
	Z80/Z8000 Commands	52

CHAPTER PAGE

VI. FUTURE PLANS 54

LIST OF REFERENCES 56

APPENDICES 59

APPENDIX A. PREP SERIAL I/O CONFIGURATION 60

APPENDIX B. PREP PARALLEL I/O CONFIGURATION 62

APPENDIX C. PREP JUMPER CONFIGURATION 65

APPENDIX D. PREP COMMAND FORMATS. 67

APPENDIX E. PREP COMPRESSION FORMATS. 72

VITA 74

LIST OF TABLES

TABLE		PAGE
1.	VERSATEC ASCII Character Set	14
2.	PREP Commands	49
3.	PREP Print Line Terminators	49
A-1.	Serial I/O Installation	60
A-2.	Serial I/O Jumpers	60
A-3.	Serial I/O Cable Connections	61
A-4.	Serial I/O Address Control	61
B-1.	Parallel I/O Installation	62
B-2.	Parallel I/O Address Control	62
B-3.	Parallel I/O Port Configuration	63
B-4.	Parallel I/O Cable Connections	64
B-5.	Parallel I/O Jumpers	64
C-1.	Interrupt Jumpers	65
C-2.	MULTIBUS Jumpers	65
C-3.	Other Jumpers	66

LIST OF FIGURES

FIGURE		PAGE
1.	Generation of Raster Plot Data.	2
2.	Sample Print Output	3
3.	Sample Plot Output.	4
4.	PREP Functional Diagram	7
5.	MSC8004 Microcomputer	10
6.	Principles of Electrostatic Writing	15
7.	PREP Block Diagram.	17
8.	8251 Mode Control Format.	18
9.	8251 Command Instruction Format	20
10.	8253 Timer Mode Select Format	21
11.	Serial I/O Initialization Sequence.	22
12.	8255 Bit Control Format	24
13.	8255 Mode Control Format.	25
14.	VERSATEC I/O Signals.	26
15.	Memory Location 0D7H Control Format	29
16.	Serial Bus Configuration.	31
17.	DDCMP Message Format.	35
18.	CRC Generation.	39
19.	PREP RECEIVE Routine.	40
20.	PREP TRANSMIT Routine	41
21.	PREP Buffer Structure	43
22.	FIFO Buffer Storage	44
23.	Determination of Transmitted Buffer	46

FIGURE PAGE

24. PREP Data Message 48

25. PREP Processing of ASCII Print Data 50

CHAPTER I

INTRODUCTION

ORNL Word Processing

At the Oak Ridge National Laboratory (ORNL) many secretaries and programmers are using minicomputers for word processing. With powerful full screen text editors, users can easily create technical documentation with complex tables and equations. Since special symbols, logos, and multiple character styles are not available on all printers, text is formatted into plot data.

Multiple users performing word processing tasks simultaneously often causes computer response to be slow. This is because creating plot data takes a large amount of time and memory. One technique of generating plot data is shown in Figure 1. Any symbol or character can be represented as many lines. Each line is comprised of points called raster units. A figure is formed by filling in specified points. If filled in points are given the value of 1 and others the value of 0, then raster plot data is obtained. Obviously, the more points used the higher quality the plot. The trade-off is more computer storage and time to process the data.

At the Oak Ridge facility, proof copies are formed inexpensively on raster plotters. These devices provide quick print (Figure 2) and plot (Figure 3) output. Final publication quality output is routed to the more costly and time consuming phototypesetter. Presently all plotting devices are located some distance from most users. The user often waits several hours for an operator to retrieve the output, separate it from other

ORNL–DWG 81-20878

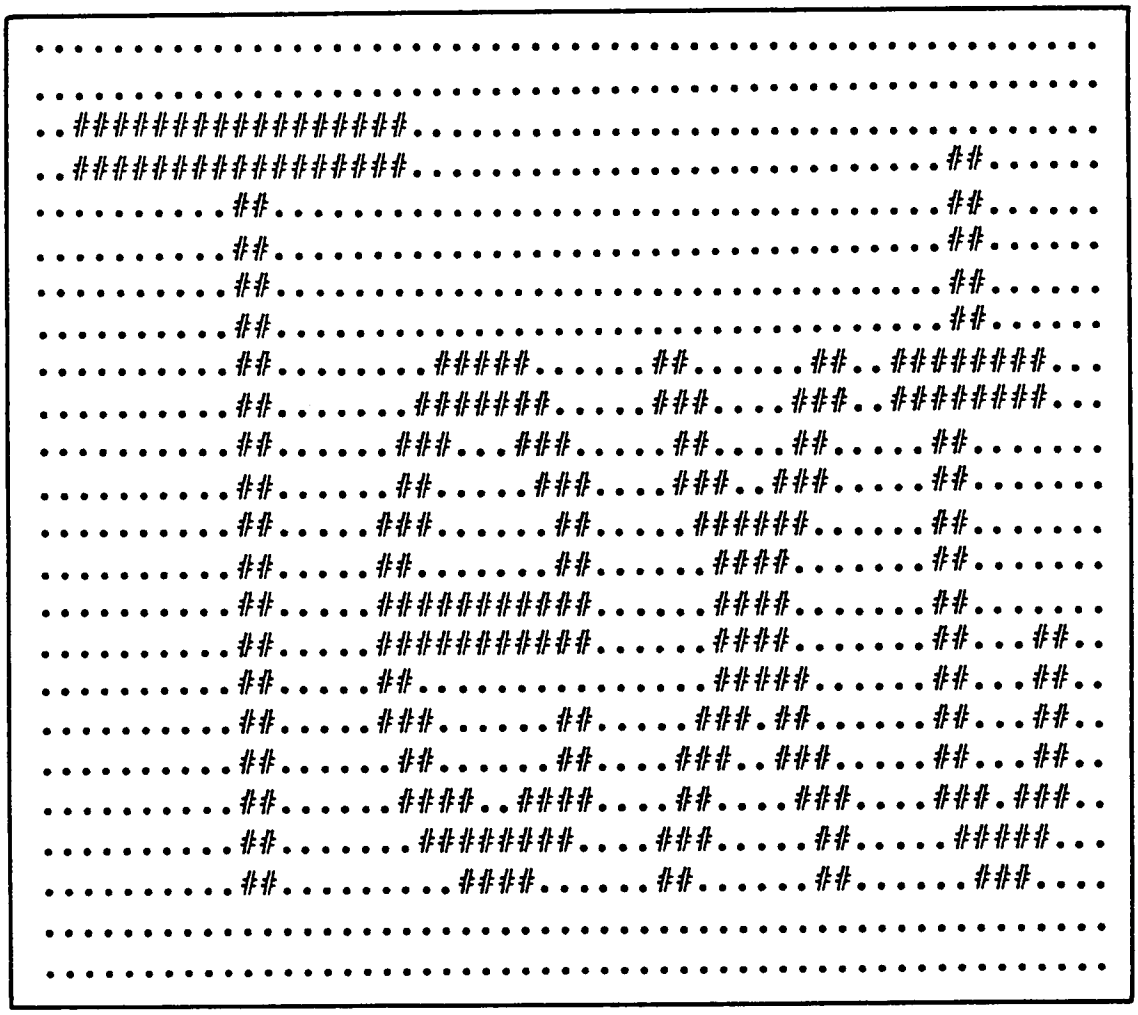


Figure 1. Generation of Raster Plot Data.

```
.SP 15
.EQ
delim $$
.EN
Much of the formal development of the Hamiltonian formulation- conservation
theorems, Poissons brackets, etc.- can easily be recast in a form suitable
for continuous systems.  Thus the modified Hamilton's priniciple becomes
.KS 2
.EQ

delta~ { int from 1 to 2 int int int { left { sum from k { pi sub k
eta dot sub k - H } right ) } dx sub 1 dx sub 2 dx sub 3 dt ~ = ~ Ø. )

.EN
.KE

As an example of the conservation theorems, let us investigate the conditions
under which the Hamiltonian is a constant of the motion.  The total time
derivative of $ H $ is

.KS 2
.EQ

dH over dt~ =~ int int int left {~ sum from k left ( {{ partial H } over
{ partial eta sub k }}~ eta dot sub k ~+~ {{ partial H } over
{ partial pi sub k)) pi dot sub k  right ) ~+~ {partial H }
over {partial t} right )~ dx sub 1 dx sub 2 dx sub 3 ,

.EN
.KE

.KS 2
.EQ

H~=~ bold pi cdot bold eta dot ~ -~ L ~=~ bold pi sup 2 over {2 mu sub Ø)
~+~ { P sub Ø gamma} over 2 { left ( del cdot bold eta right )} sup 2 .

.EN
.KE

.KS 2
.EQ

  { partial H } over { partial eta sub k }~ - ~

  sum from j ~{d~) over {dx sub j}

  left { {partial H } over { {partial { left { {partial eta sub k } over

{partial x sub j} right ) } } } right )~ =~ - pi dot sub k ,

.EN
.KE
```

Figure 2. Sample Print Output.

Note: This test contains the word processing commands used to create Figure 3.

Much of the formal development of the Hamiltonian formulation- conservation theorems, Poissons brackets, etc.- can easily be recast in a form suitable for continuous systems. Thus the modified Hamilton's priniciple becomes

$$\delta \int_1^2 \int\int\int \left\{ \sum_k \pi_k \dot{\eta}_k - H \right\} dx_1 dx_2 dx_3 dt = 0.$$

As an example of the conservation theorems, let us investigate the conditions under which the Hamiltonian is a constant of the motion. The total time derivative of H is

$$\frac{dH}{dt} = \int\int\int \left\{ \sum_k \left(\frac{\partial H}{\partial \eta_k} \dot{\eta}_k + \frac{\partial H}{\partial \pi_k} \dot{\pi}_k \right) + \frac{\partial H}{\partial t} \right\} dx_1 dx_2 dx_3,$$

$$H = \pi \cdot \dot{\eta} - L = \frac{\pi^2}{2\mu_0} + \frac{P_0 \gamma}{2} \left(\nabla \cdot \eta \right)^2.$$

$$\frac{\partial H}{\partial \eta_k} - \sum_j \frac{d}{dx_j} \left(\frac{\partial H}{\partial \left(\frac{\partial \eta_k}{\partial x_j} \right)} \right) = -\dot{\pi}_k,$$

Figure 3. Sample Plot Output.

output, and distribute it. In many cases "turn-around-time" includes courier delivery or mailing.

To make word processing at the Oak Ridge National Laboratory more user efficient, a device was proposed that could provide plotting stations at remote locations and execute word processing tasks. This enhancement to the system would greatly improve user response on the host computer and provide a quick, accessible copy of output.

Possible Enhancements

A survey of available devices on the market revealed one controller that could support a remote plotter. The controller is built by a manufacturer of plotters and does provide a quality hardcopy of ASCII character print data and raster plot data.

Host compatible software available from the same vendor is used to compress plot data. Compression of raster data is accomplished by rewriting the data in descriptive terms. For instance, a string of ones could be compressed by simply storing the number of repeats. Using compressed plot data requires less storage and allows faster transmission of the plot from one location to another.

Print or compressed plot data is transmitted from the host to a microcomputer. The microcomputer decompresses the plot data and sends the print data or uncompressed plot data to the plotter. Although this device does provide efficient remote plotting capabilities, the flexibility of modifying software, adding a different plotter, and executing existing word processing programs is not feasible.

Another potential enhancement involved the work of a group of computer scientists and engineers at Stanford University. They programmed a

microcomputer to receive data from a host, transform it to elaborate character fonts, and then transmit it to the plotter. For ORNL's purposes, this device did not provide remote plotting and was much too task specific.

The PREP Controller

A combination of the features of the above devices and new ones, resulted in a more general controller. The programmable remote plotting device (PREP) consists of two microcomputers and a raster scan plotter.

One microcomputer is used as the master of PREP. It performs the following tasks:

1. Receive ASCII characters or raster data from the host.
2. Transmit ASCII characters or raster data to the plotter.
3. Transmit status information from the controller to the host.
4. Parse and execute commands from the host.
5. Expand compressed raster data.
6. Download word processing programs from the host.

The other microcomputer is controlled by, or slaved to, the master. It is a more powerful computer and is used primarily to execute word processing tasks.

Figure 4 is a circular diagram depicting PREP's functional structure. Each slice of the circle represents the interaction of the host computer, the master microcomputer, the plotter, and the slave microcomputer. Beginning at the top of the circle, the master receives and transmits information to the host computer. If print data or uncompressed raster data is received, the master transmits it directly to the plotter. Compressed raster data is expanded by the master before being plotted. Large word processing programs are stored and executed by the slave

ORNL-DWG 81-20879

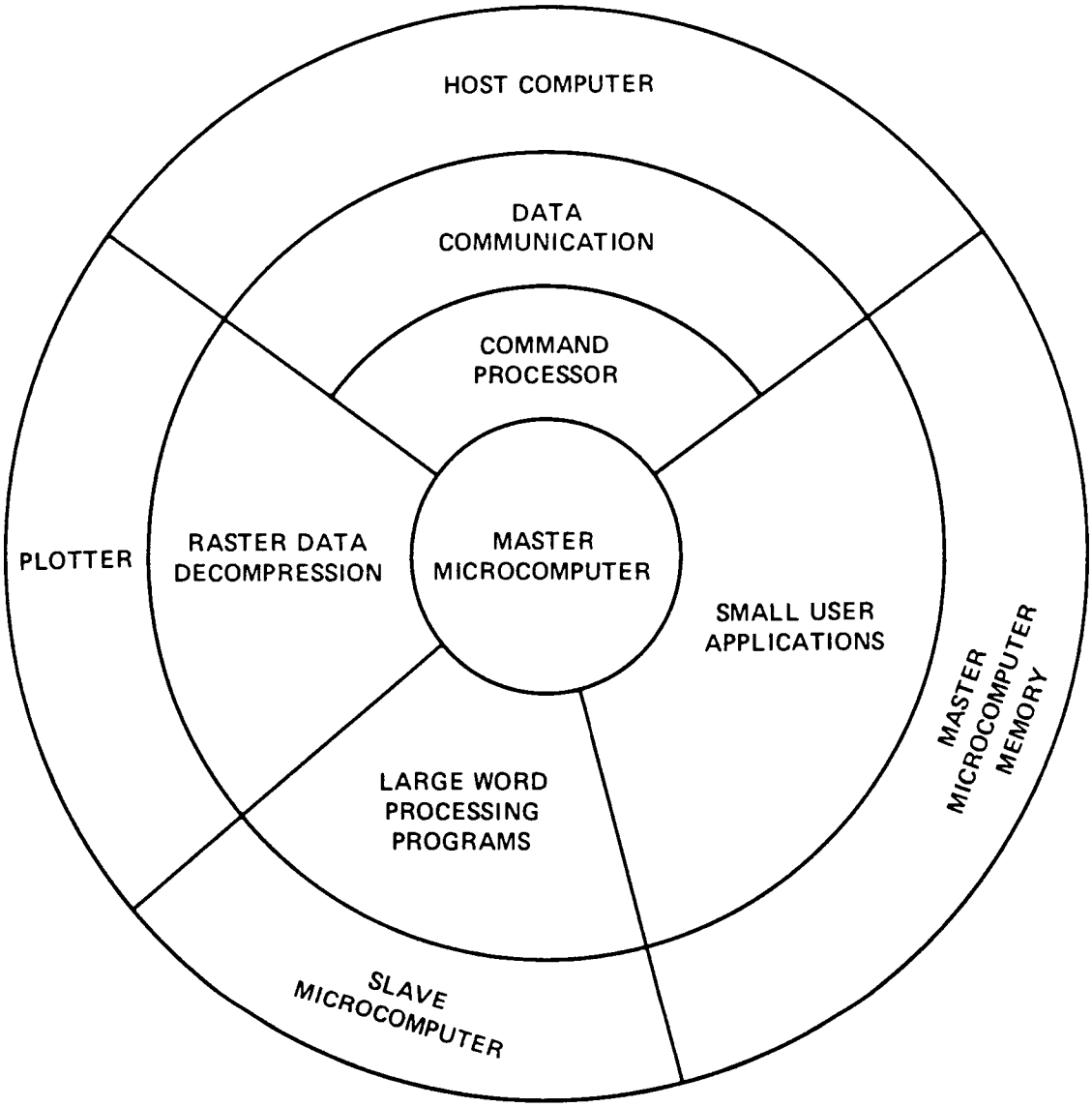


Figure 4. PREP Functional Diagram.

microcomputer. Smaller tasks or revisions to the controller are stored and utilized by the master microcomputer.

An attractive feature of PREP is its task flexibility. By transferring computer code, the host can dynamically program the controller to execute new tasks that are currently overloading it. Modifications to existing tasks can also be transmitted.

The PREP hardware design is also very flexible. The controller is easily interfaced with a variety of host computers. Different plotting devices can be added with very few modifications. The PREP controller is designed so that it can operate without the slave microcomputer. This modification does not allow large task processing but does cut the cost of the controller in half.

CHAPTER II

COMPONENT DESCRIPTIONS

Host Computer

The host computer is the Digital Equipment Corporation PDP 11/70. This minicomputer is the most powerful of Digital's family of 11s. It is classified as a medium sized computer and is designed to operate in large multiuser, time-sharing systems.

The 11/70s at ORNL are run with UNIX operating systems. The UNIX system provides file management and text processing for up to 40 simultaneous users. Programming languages available are C (a high level language conducive to structured programming), BASIC, and FORTRAN.

Currently three 11/70s support over 200 users. Future plans include adding more users and minicomputers to the system.

Master Microcomputer

All PREP operations are controlled by a single board Monolithic Systems Corporation microcomputer. As shown in Figure 5, the MSC8004 is driven by a ZILOG Z80 microprocessor.

The Z80 has 16 general purpose registers. Special purpose registers include stack pointer, program counter, interrupt vector, memory refresh, and index registers. The Z80 assembly language executes 8- or 16-bit loading, arithmetic and logical manipulations, register exchange, block transfer and search, rotate and shift, bit setting and testing, jumps, subroutines, and input/output instructions.

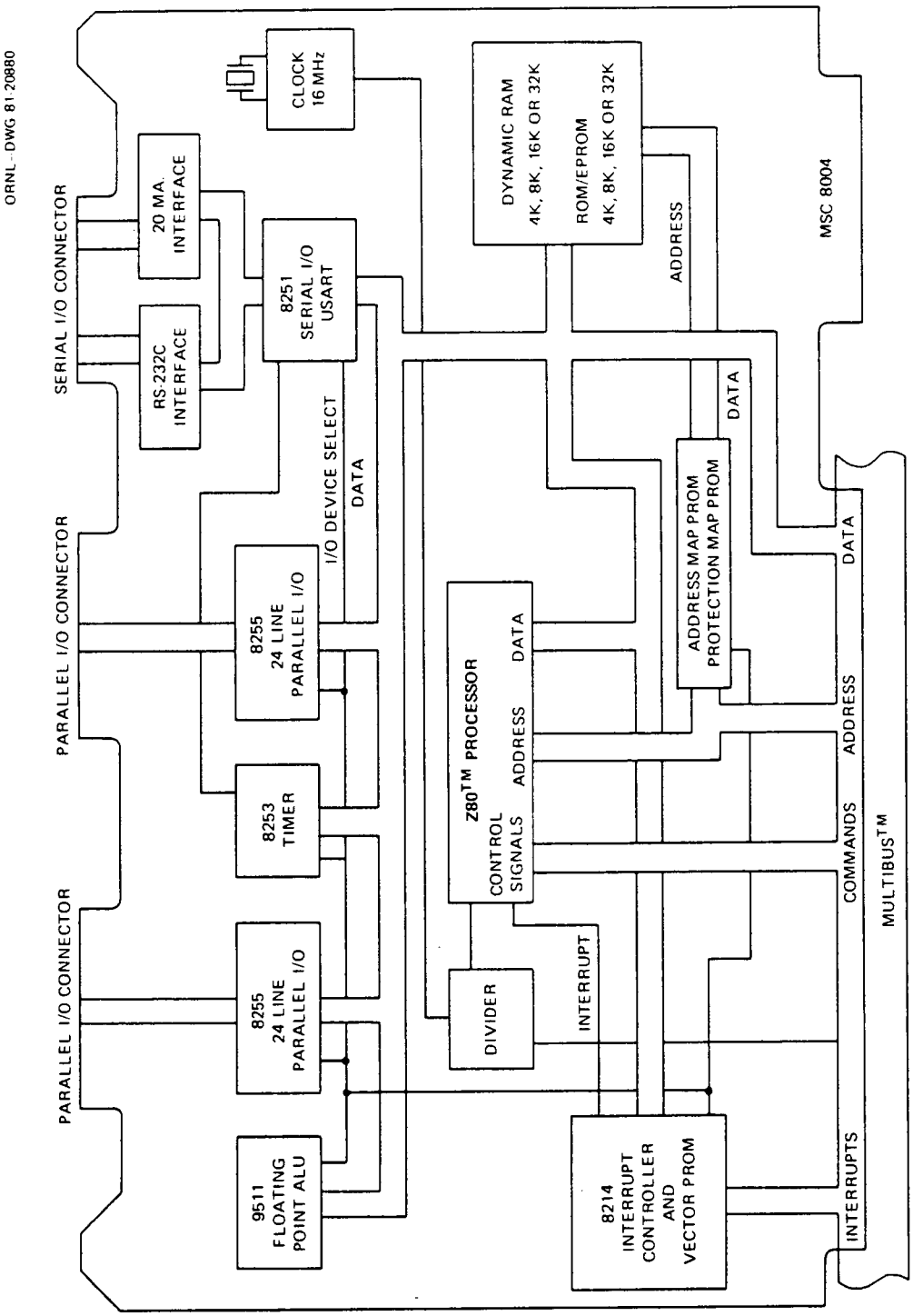


Figure 5. MSC8004 Microcomputer.

The MSC8004 can be configured with up to 32K bytes of both read only memory (ROM) and dynamic read/write memory (RAM). The microcomputer is also capable of handling both serial and parallel I/O.

Two programmable chips provide serial data communication. The 8251 USART (Universal Synchronous/Asynchronous Receiver/Transmitter) can configure the board for RS-232C, TTL, or opto-isolated 20 ma current loop. The 8253 programmable timer supplies the transmit and receive clock. Thus by programming these devices, the user can select data format, control character format, parity, and baud rate.

Two 8255 programmable peripheral interface controllers receive and transmit parallel data. These devices provide the user with three modes of operating 48 parallel lines. These modes are six 8-bit ports, four 8-bit ports with handshaking, and bidirectional port control.

The 8214 interrupt controller and vector prom provide eight levels of priority and non-maskable interrupts. The 8214 can be programmed to transfer control to fixed memory locations, or with the aid of the vector prom, transfer to any memory location desired.

An internal bus is used to send address, data, or control signals between the Z80 processor, memory, I/O or other devices. The MSC8004 is also equipped with control logic to make it MULTIBUS compatible. MULTIBUS, introduced by INTEL for single board computers and standardized by IEEE 796, is used to interface with 8- and 16-bit computers, memory boards, and peripheral controllers.

Slave Microcomputer

The slave microcomputer is a COMPUTEX Z8000. It has MULTIBUS compatible CPU and memory boards. The CPU board uses a ZILOG Z8001 16-bit microprocessor chip as the computer's main processor.

The Z8000 is a very powerful processor. In typical applications, it is 5 to 10 times faster than any 8-bit microprocessor (including Z80), and is 2 to 5 times faster than any 16-bit microprocessor. The Z8000 has 16 general purpose registers, many data element types, a large instruction set, and eight addressing modes. It can perform 32-bit operations in one instruction, string manipulation, block I/O, and has powerful interrupt handling.

Two programmable chips are used on the CPU board to control interrupt processing. The 8253 timer can be programmed to set timers to generate interrupts. The 8259A interrupt controller allows eight priority interrupt levels to be set. Besides fixed priorities, the 8259 allows software polling to determine what service is to be performed. It also assigns equal priority where each level is rotated to the lowest level after an interrupt.

Also included on the board is a AMD 9511 arithmetic processing unit. It performs complex floating point arithmetic operations including addition, subtraction, multiplication, and division. Floating point operands can also be used with logarithmic, trigonometric, and other functions.

Although the Z8000 can directly address up to 8 megabytes of memory, the microcomputer is configured with 128K RAM. A memory management circuit

on the board divides memory in 64K byte segments. A segment is allocated in 2K byte pages. Either 8- or 16-bit transfers can be performed.

Plotter

The output device used by the PREP controller is a VERSATEC V-80 electrostatic printer/plotter. Simple loading of paper and ink and relatively quiet operation make the V-80 easy to use in an office environment.

The V-80 can print ASCII characters (see Table 1), plot raster data, or simultaneously do both. Each process is accomplished by electrostatic writing. Figure 6 shows the principles of this writing operation.

Through microprocessor control, the V-80 places an electrostatic charge on specified locations or nibs on the writing head. As the paper passes over the nibs, the charges are transferred onto the paper. The paper is then exposed to a special ink or toner. Black particles from the toner adhere only to the charged surfaces of the paper. Excess toner is vacuumed off so that the resulting paper is dry and ready to use.

The V-80 has a high resolution output of 200 dots per inch. It prints at a maximum rate of 1000 lines per minute and plots at 1 inch per second.

Data transmission to the printer/plotter can be accomplished through a parallel interface. The data are then stored in internal buffers. The writing nibs (2112 per line) correspond to each bit of incoming data and are charged when line termination occurs or a full line of data has been stored. A full plot line consists of 264 bytes. A print line is 132 characters with each character spanning 16 nibs.

TABLE 1

VERSATEC ASCII Character Set

BIT POSITIONS 3-0	BIT POSITIONS 6-4							
	0	1	2	3	4	5	6	7
0			SP	0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4	EOT		$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT		+	;	K	[	k	{
C	FF		,	<	L		l	|
D			-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

Note: SUB will be printed as a backward ?.

ORNL–DWG 81-20881

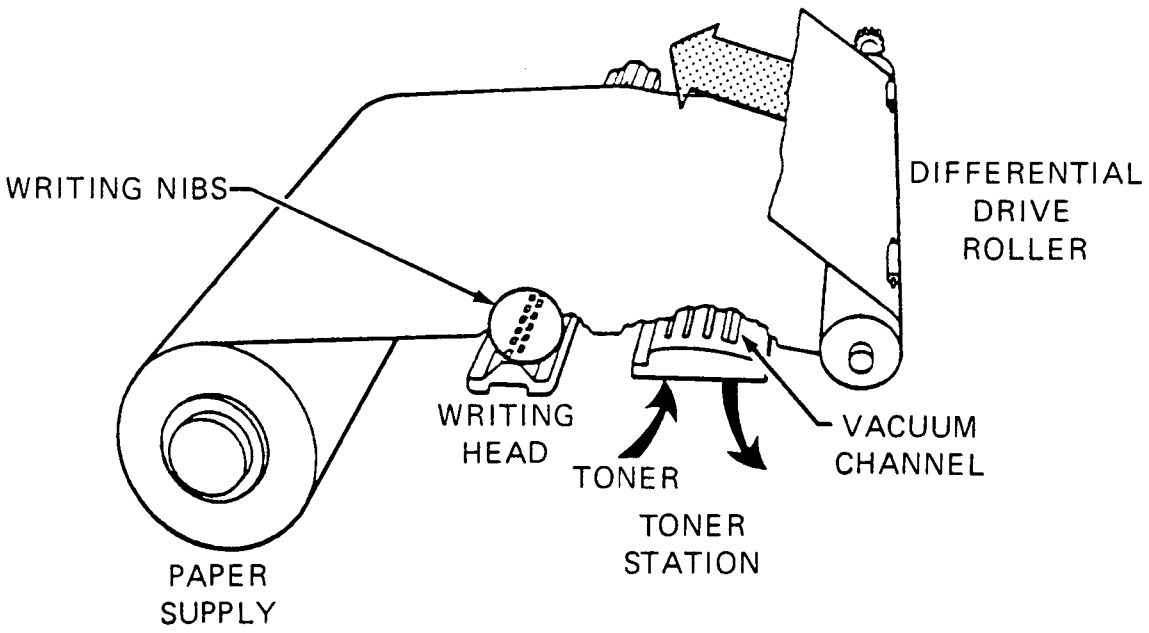


Figure 6. Principles of Electrostatic Writing.

CHAPTER III

SYSTEM CONFIGURATION

Figure 7 depicts a simplified block diagram of PREP. As shown, the MSC8004 is used to interface with each component. It communicates with the host 11/70 computer through a serial I/O port. It uses parallel I/O ports to transfer information to the VERSATEC plotter. It accesses the Z8000 computer through MULTIBUS. The MSC8004 also provides interrupt capabilities.

Because the MSC8004 has empty I/O sockets and no jumper connections, each of the above interfaces requires hardware installation and configuration. Many of the powerful chips used such as the 8251, 8255, and 8214 also require programming to select the desired operation. The following is the system configuration designed for the PREP controller.

Serial I/O Port

PREP communicates with the host computer by transmitting and receiving serial data. The 8251A USART, the 8253 timer, driver and receiver chips configure the MSC8004 with a serial I/O port. These components and their corresponding board destinations are described in Appendix A-1. An EIA RS-232 interface is used to link the microcomputer with the host computer. Appendix A-2 supplies the jumper pin connections necessary for the RS-232 configuration. Appendix A-3 contains the serial I/O cable connection and Appendix A-4 gives MSC8004 addresses used to control the 8251 and 8253.

Both the 8251A and the 8253 must be programmed to initiate data communications. The 8251 is controlled by using a command or mode instruction. The mode control instruction format is four 2-bit fields

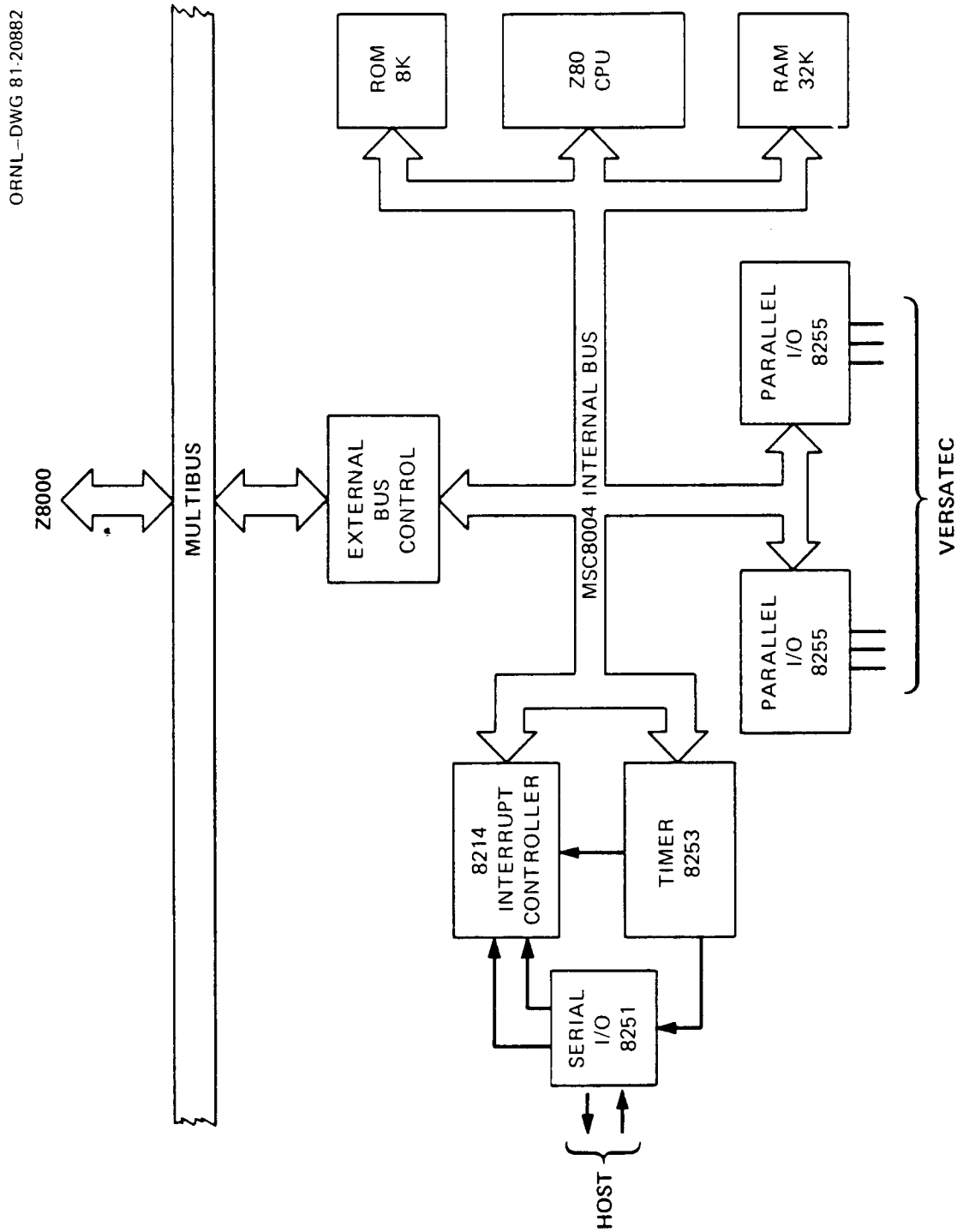


Figure 7. PREP Block Diagram.

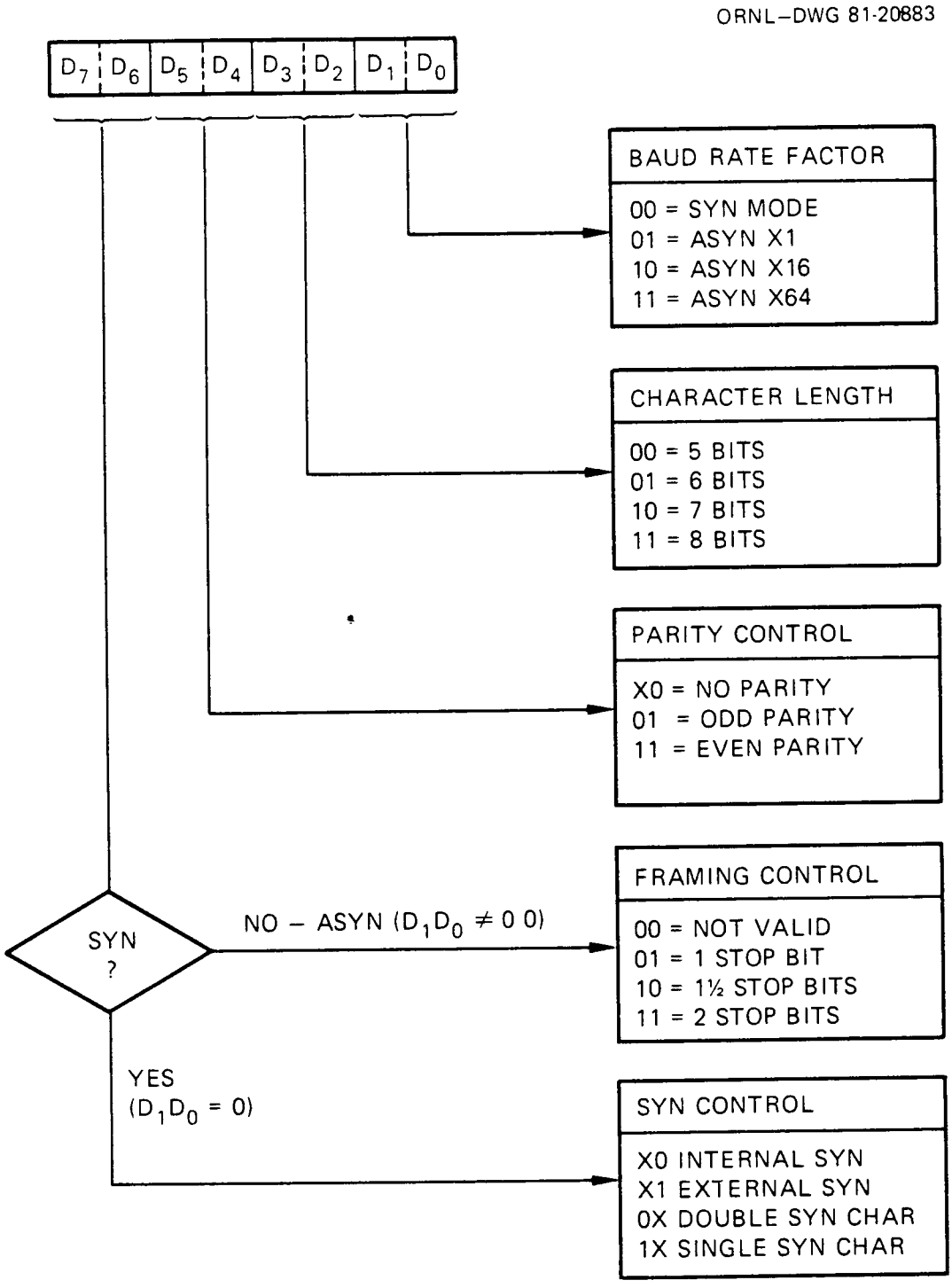


Figure 8. 8251 Mode Control Format.

(Figure 8). These fields determine synchronous or asynchronous operation, baud rate factor, character length, parity, number of stop bits or number of sync characters. The command instruction format is eight 1 bit fields (Figure 9). This instruction enables data communications, resets error flags, controls modem signals, and returns the 8251 to mode control.

The 8253 timer is used to generate clock pulses that control the rate of character transmission. The timer works by loading a counter and emitting a pulse after decrementing the counter to 0. The 8253 is initialized by loading a mode select word followed by one or two count values. This word (Figure 10) is used to select one of three timers, the number of bits (8 or 16) used as the counter value, the type of countdown, and the type of pulse generated.

Figure 11 depicts the sequence necessary for initialization of the individual chips. After the 8253 has been set, the 8251 is reset. Resetting occurs when the 8251 is loaded with three consecutive zeroes followed by a single value of 40H. After the mode and command instructions have been loaded, the USART can now receive and transmit data. To avoid the problem of extraneous data in the USART at reset time, a dummy read is performed before data communications begin. To insure against bad data, a short delay is used between successive reads. This technique is employed because the CPU operates twice as fast as the 8251 and might access data before it has stabilized.

The 8251 used in PREP has been configured to receive and transmit 8-bit characters with one stop bit and no parity checking. It operates in asynchronous mode. A 9600 bits per second baud rate is generated by assigning values to both the 8251 and 8253. The 8253 timer No. 2 is loaded

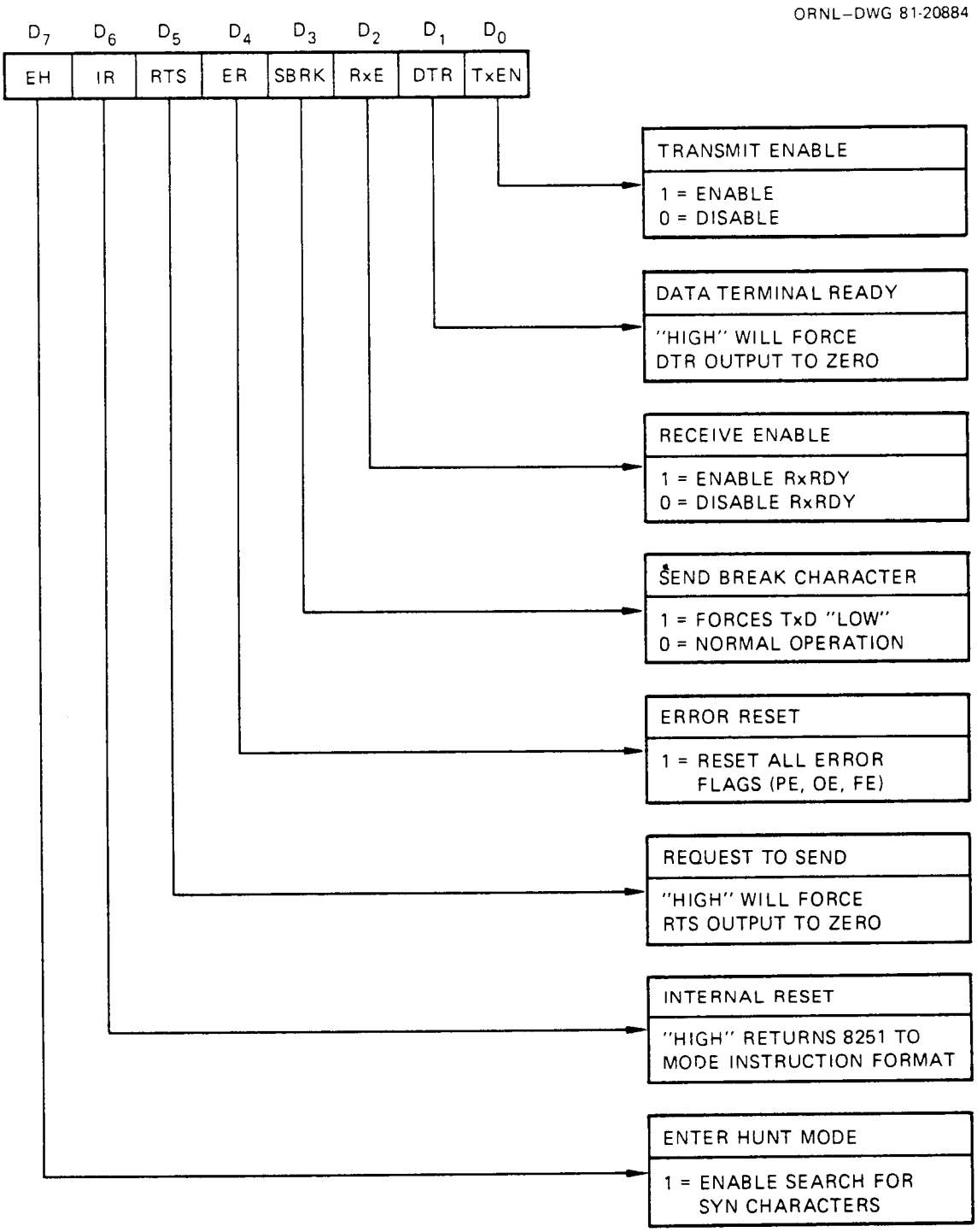


Figure 9. 8251 Command Instruction Format.

ORNL–DWG 81-20885

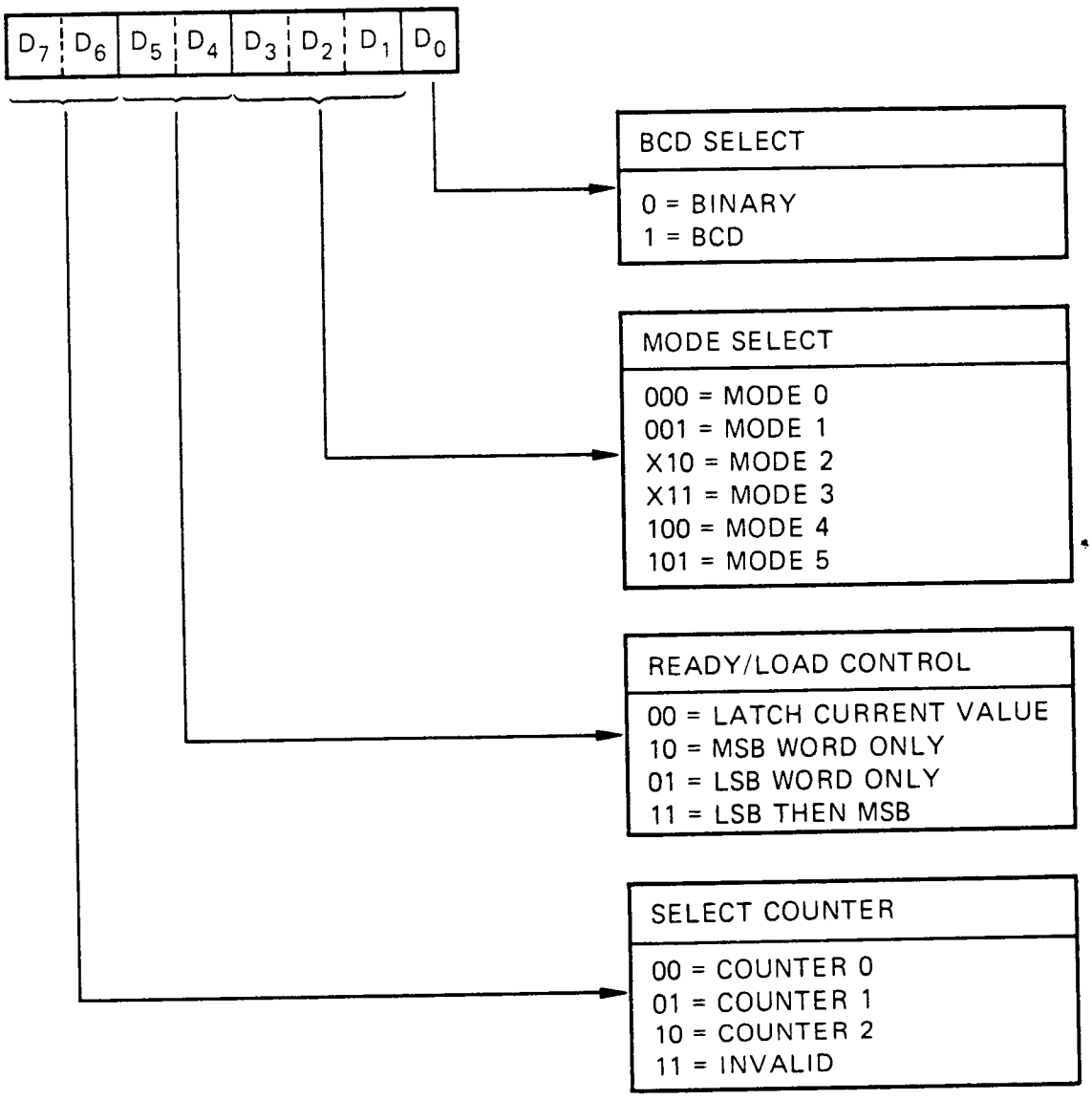


Figure 10. 8253 Timer Mode Select Format.

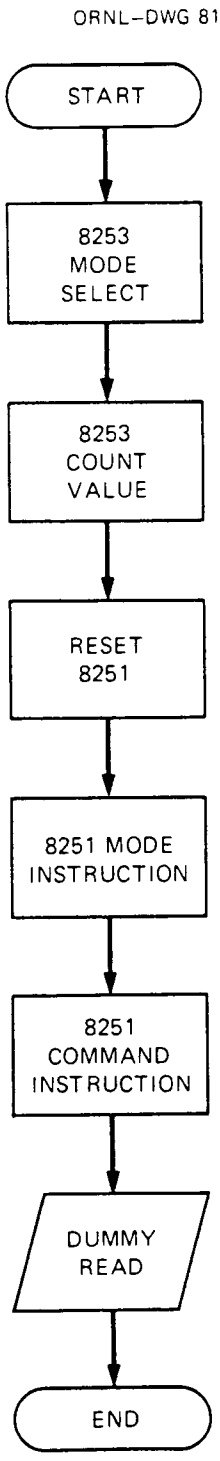


Figure 11. Serial I/O Initialization Sequence.

with 13 so that the 2MHz driven device outputs a square wave (Mode 3) with a 2000000/13 frequency. This pulse is fed to the 8251 whose baud rate factor is 16. The baud rate factor causes the frequency to be divided by 16, resulting in a frequency which is approximately 9600 baud.

Parallel I/O Ports

PREP communicates with the plotter through parallel communication. Six 4.7K pull-up resistors, two hex buffers, four octal transceivers, and two 8255 peripheral interface chips are installed to configure the MSC8004 with a parallel interface. These chips provide up to three parallel I/O ports at the J1 and J2 edge connectors on the board. Appendix B-1 lists the board destinations for these components. Appendix B-2 gives the MSC8004 addresses necessary to control, read, or write to ports A, B, and C.

The 8255 chips are programmed to initialize the parallel ports. The 8255 control word register is used to specify operating mode for all ports and to initialize Port C. The Bit Control Word format (Figure 12) is used to set or reset any of the eight bits of Port C. After this initialization, the Mode Control Word is written. Its format (Figure 13) consists of 2 groups. Group A is used to select the mode and functions for Port A and the upper 4 bits of Port C. Group B denotes these functions for Port B and the lower 4 bits of Port C.

The 8255 ports are used to transmit and receive plotter data signals. As shown in Figure 14, the VERSATEC V-80 requires 20 signals to handle data transfer, status information, mode control, and remote commands. The plotter receives 8-bit bytes of either raster or ASCII data on lines IN01-IN08. The READY output signal indicates when the plotter can accept

ORNL–DWG 81-20887

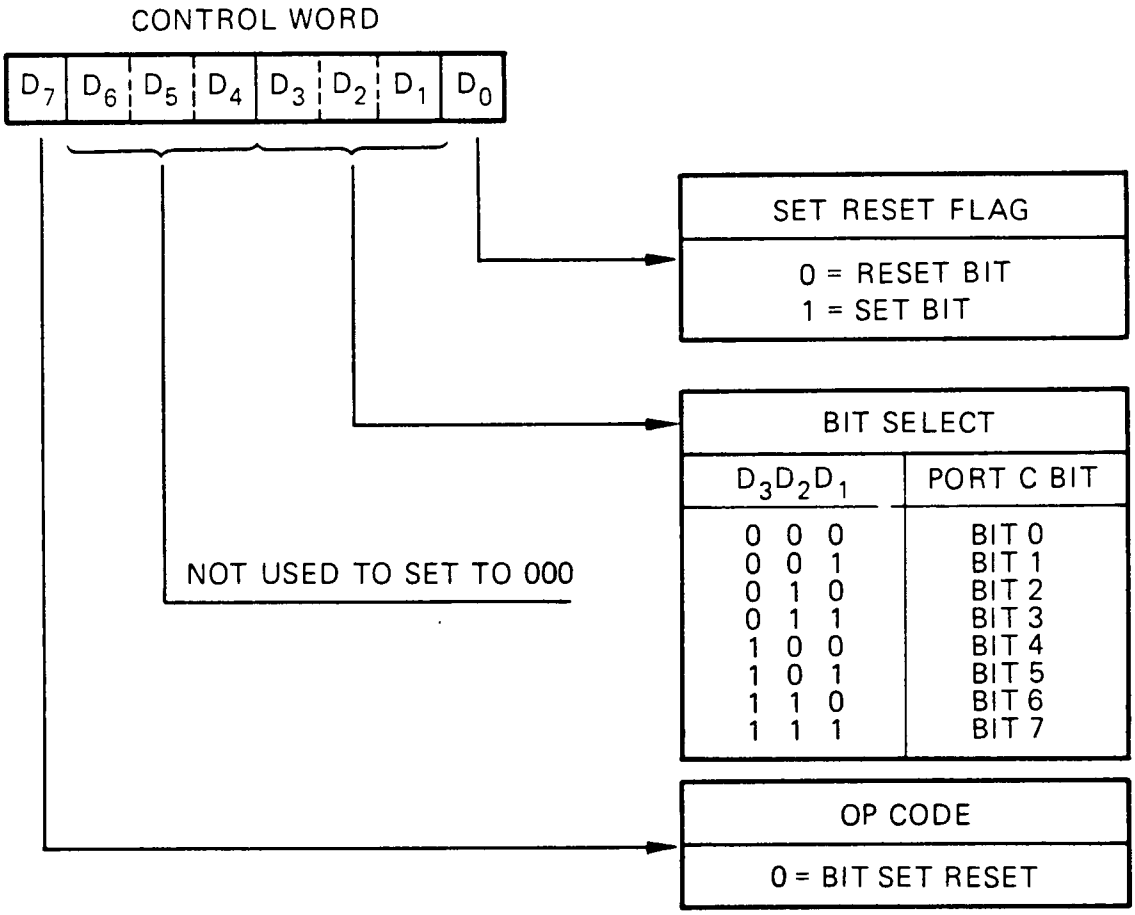


Figure 12. 8255 Bit Control Format.

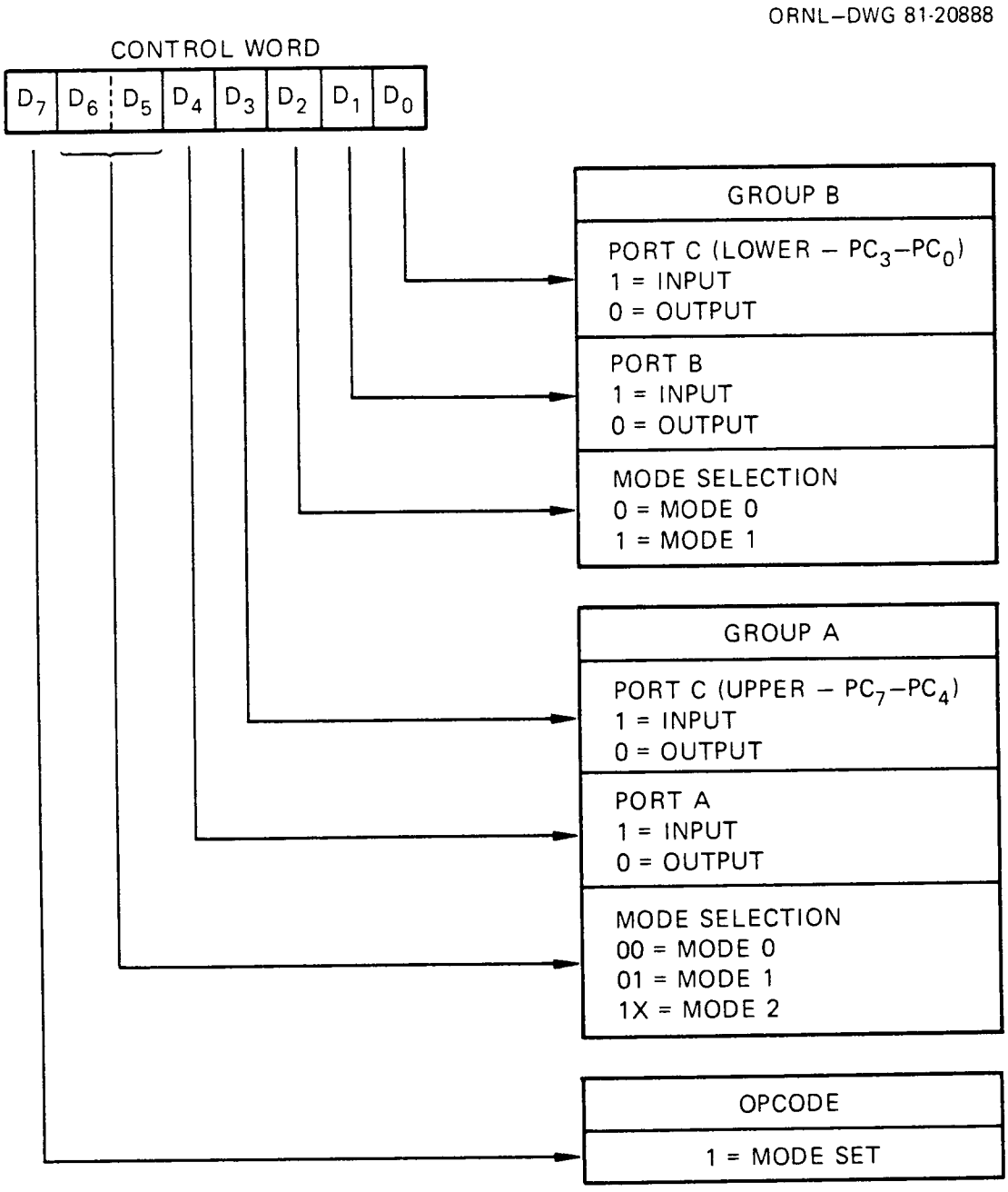


Figure 13. 8255 Mode Control Format.

ORNL–DWG 81-20889

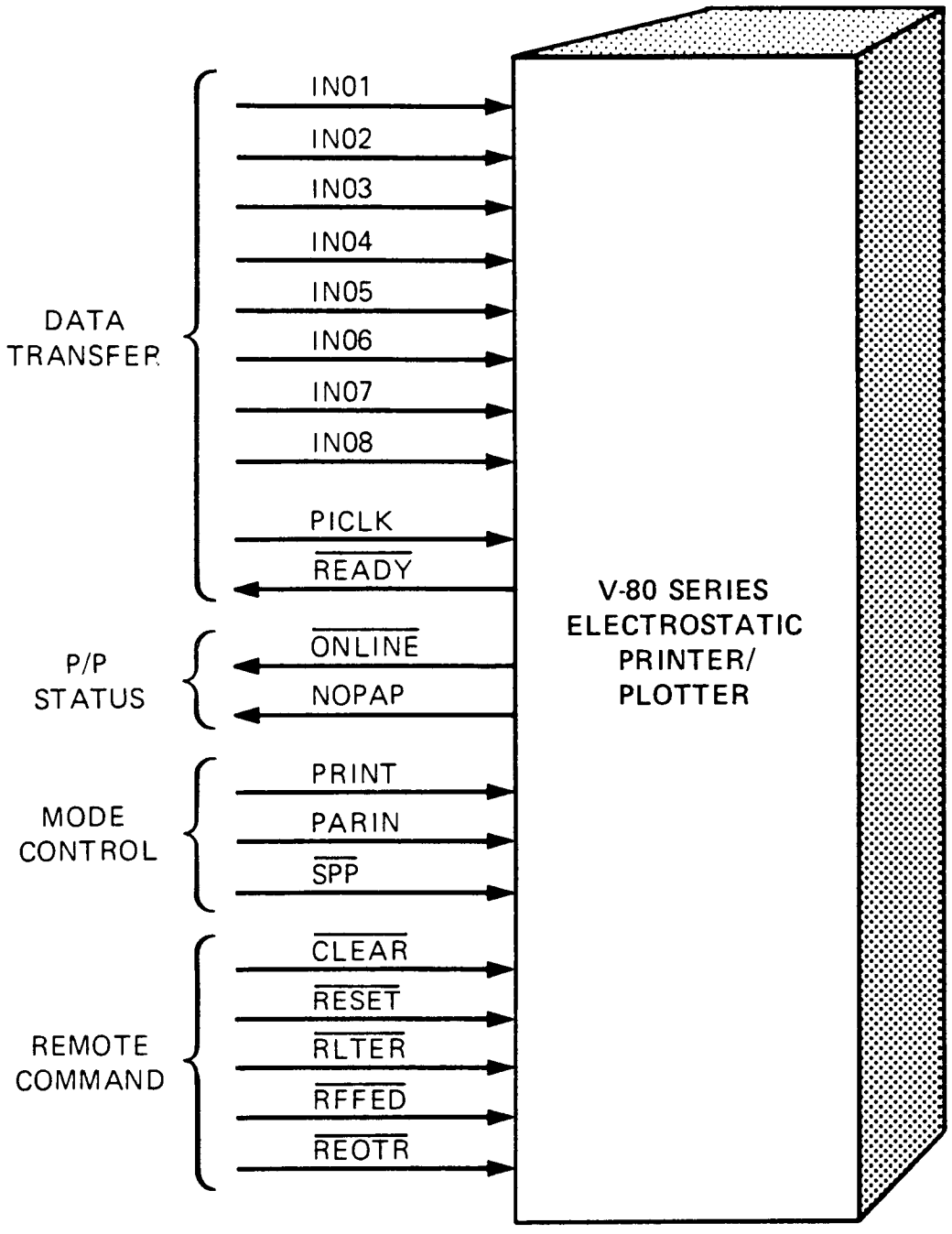


Figure 14. VERSATEC I/O Signals.

incoming data or commands. Information is then clocked into the plotter according to the PICLK pulse.

Three control signals are used to select the VERSATEC's operating mode. PRINT tells the plotter to print or plot the incoming data. PARIN selects parallel or serial input. SPP places the plotter in a simultaneous print/plot mode.

The plotter has five command signals. CLEAR is used to zero the plotter input buffer. RESET initializes all internal circuitry and puts the machine in print mode. Print or plot lines are terminated by RLTER. A form feed occurs with RFFED. Activation of the Remote End Of TRansmission (REOTR) causes two form feeds.

In addition to READY, the plotter outputs two other status signals. ONLIN simply denotes when the plotter is on. NOPAP indicates a low supply of either paper or ink.

Appendix B-3 shows how each port is configured. The J1-Port A is used to transmit the remote commands. Print or plot data are sent to the plotter on J1-Port B. J1-Port C supplies the PICLK pulse. It is generated using the handshaking signals provided by Mode 1 operation. A low signal on Bit 1 of Port C indicates a Output Buffer Full (OBF) condition. This signal denotes data transfer from the Z80 CPU to Port B has occurred. Bit 2 is a low true acknowledgement (ACK) to the 8255 that data has been received by the peripheral device. Connecting the OBF signal to the ACK signal causes the data to be loaded and acknowledged simultaneously. The resulting strobe signal is PICLK.

All J2 ports function with basic I/O capabilities (Mode 0). J2-Port A outputs mode control. J2-Port B receives status information. J2-Port C is not used.

An interface cable is used to connect the plotter signals to the 8255 ports. Appendix B-4 gives pin assignments used on the VERSATEC and J1, J2 connectors. Jumper connections necessary for the parallel interface are given in Appendix B-5.

Interrupt Structure

The initialization of the interrupt system is accomplished through Z80 instructions and the 8214 interrupt controller. Z80 instructions are used to enable or disable interrupts, set interrupt modes, and program the vector prom. The 8214 causes interrupts to occur. It is controlled by writing to location 0D7H using the format specified in Figure 15. This format specifies one bit for interrupt mode and four bits for enabling the eight interrupt levels.

A prom is used in Mode 2 to specify interrupt vector addressing. PREP utilizes 3 interrupt levels. The highest priority level is connected to the 8251 RxRDY output. This signal goes true to indicate the 8251 has received a character. The 8251 TxRDY output is assigned to the next priority. It signals the MSC8004 that the USART has just transmitted a character.

The lowest priority is given to the output of the 8253 timer No. 1. It functions as a watchdog timer for certain PREP operations. What these operations do will be discussed later, but it should be noted that each operation requires an interrupt to occur after 1 to 60 seconds. The longest time delay a 16-bit counter driven by a 2MHz clock can generate is approximately .03 seconds. To increase this range, the MSC8004 is configured with the 8253 timer No. 0 generating a clock pulse for timer No. 1. Timer No. 0 is set to provide a continuous signal (Mode 3) to timer 1

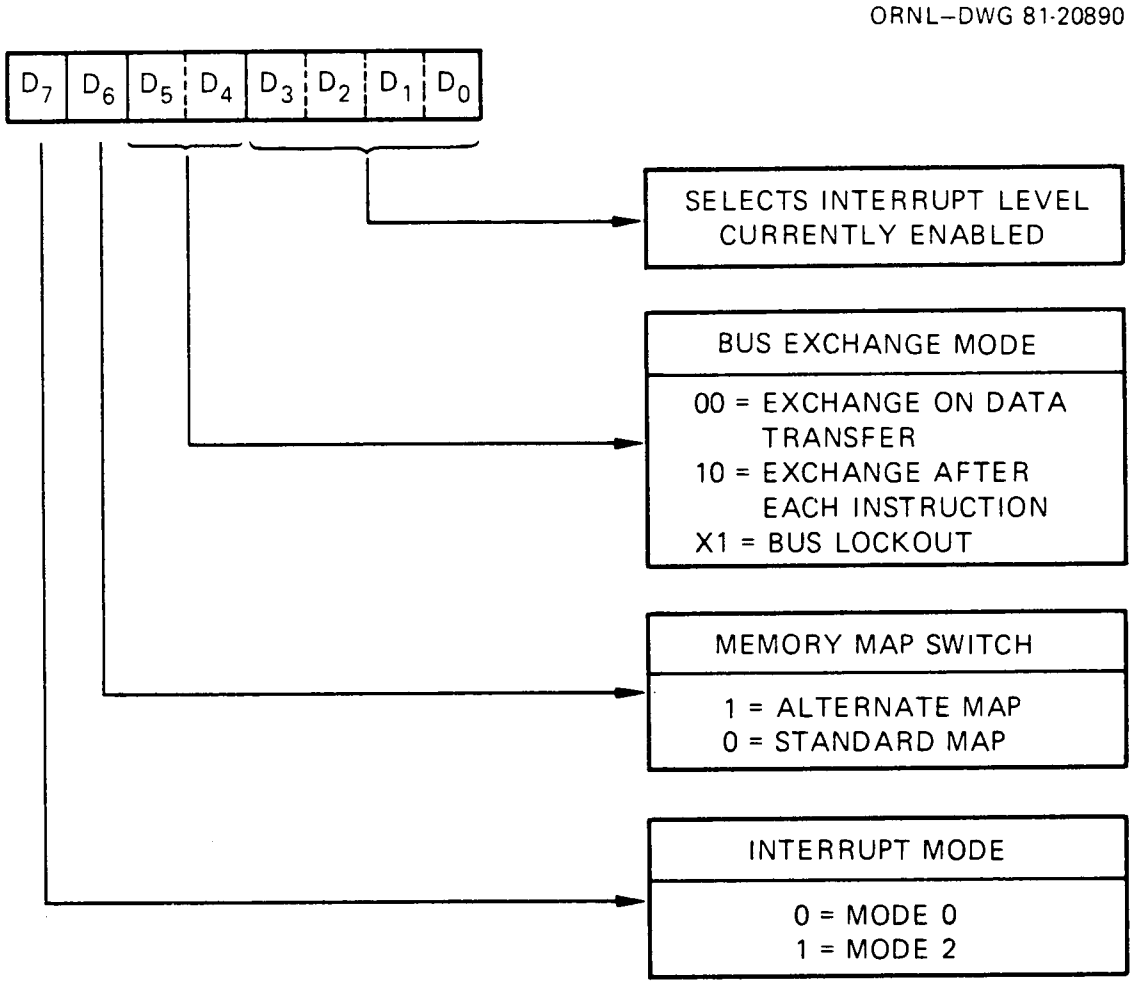


Figure 15. Memory Location 0D7H Control Format.

at a frequency of 1 millisecond. Timer No. 1 can now generate an interrupt request signal after delaying up to 65536 milliseconds.

TxRDY, RxRDY, and the output from timer No. 1 are connected to the 8214 interrupt controller by soldering or wirewrapping pins together on the board. Appendix C-1 supplies the MSC8004 pin connections.

MULTIBUS

The COMPUTEX MULTIBUS Mother Board is used to connect the MSC8004 to the COMPUTEX Z8000. The Mother Board is contained in a cabinet with a card case that holds up to 15 cards. The cabinet also supplies +5, +12, and -12 voltages to the boards. A separate -5V power supply has been added to accommodate MSC8004 requirements.

MULTIBUS contentions between the Z8000 and the MSC8004 are resolved using serial bus priority. In this configuration, the position of each module determines its priority. As shown in Figure 16, the modules are linked together by common signals. Each signal is synchronized by the bus clock (BCLK). A low on Bus PRiority iN (BPRN) establishes the master module. A high on Bus PRiority Out (BPRO) indicates to all lower priority units that the bus is not available. BPRO is generated by a logic OR of the module's interrupt request and BPRN. BUSY signifies that some module has control of the bus.

Jumper connections (Appendix C-2) must be made to the MSC8004 to support the above MULTIBUS capabilities. In addition to supplying the bus clock, these jumpers establish the MSC8004 as the master. Bits 4 and 5 of location 0D7H are set to control the bus exchange mode. PREP allows bus exchange after each instruction.

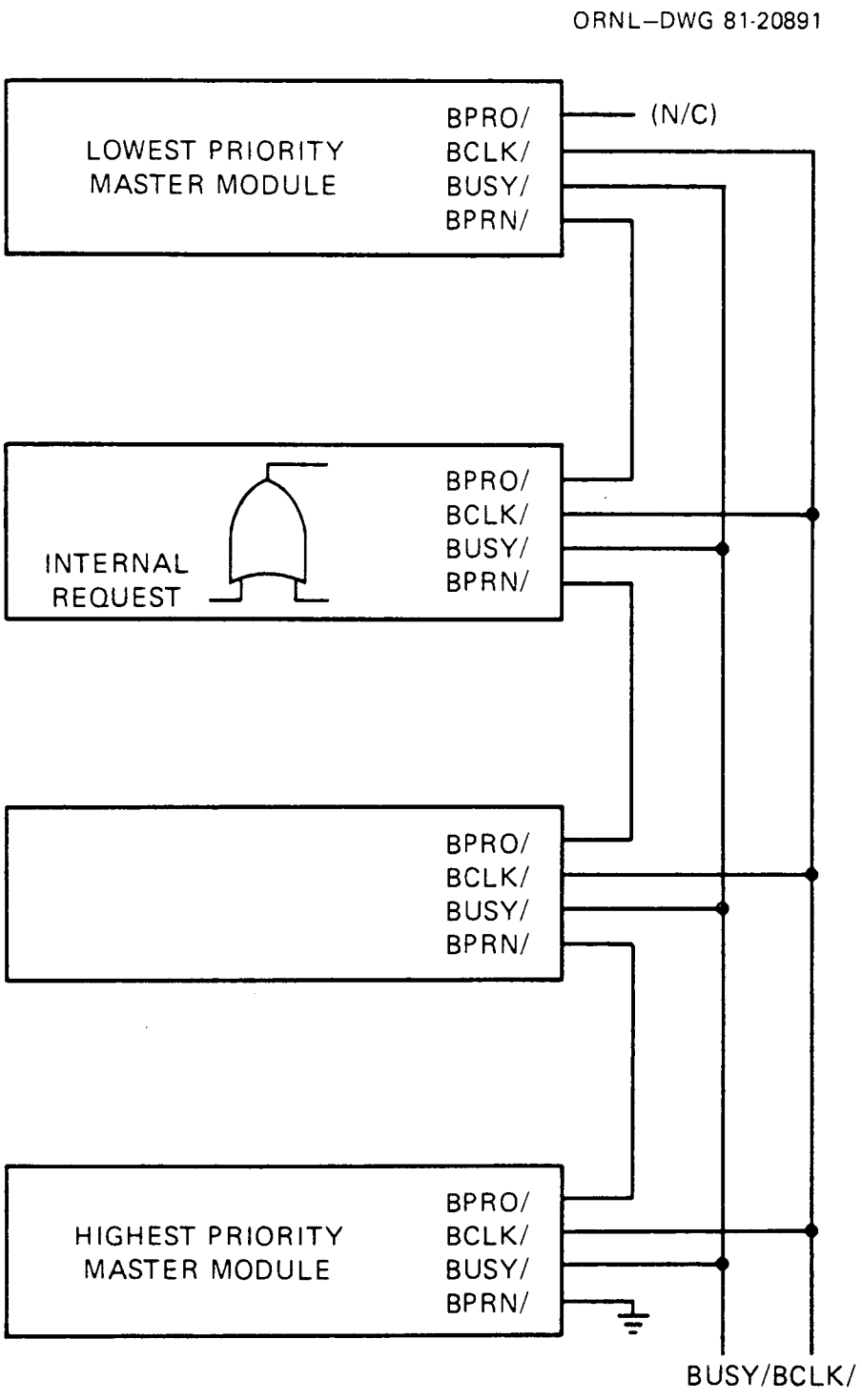


Figure 16. Serial Bus Configuration.

By removing the prom at board location U53, PREP is designed so that the Z8000 is not permitted to write into MSC8004 memory. This protects the MSC8004 memory from Z8000 programs that perform erroneous writes.

Other Connections

Bit 6 of location 0D7H determines MSC8004 address control. PREP uses the standard memory map on the board. Jumper connections not already mentioned can be found in Appendix C-3. These connections are used to supply a clock to the CPU, remove wait states, and configure the board with 32K RAM and 8K ROM.

CHAPTER IV

REMOTE DATA COMMUNICATIONS

Receiving Data

PREP receives data from the host computer using a communications protocol. This set of rules is necessary for long distance communication. It ensures information is not lost or misconstrued and defines how data is to be processed. A protocol detects transmission errors with Longitudinal, Vertical, or Cyclical checks. It groups information into numbered messages so duplicate or lost messages are detected. Handshaking procedures establish the communication of messages between two or more devices.

Three types of protocols are commonly used in data communications. They provide character oriented, bit oriented, and byte oriented messages. One of the best known character oriented protocols is IBM's Binary Synchronous protocol. BISYNC uses special characters to indicate the beginning and the end of its messages. The bit oriented protocol frames its messages with bit patterns. Examples of this type of protocol are IBM's Synchronous Data Link Control (SDLC) and International Standard Organization's High-level Data Link Control (HDLC). Byte oriented protocols use one or more bytes to denote the beginning of a message. This header includes a byte count of the number of data bytes that follow the header. Digital Equipment Corporation's Digital Data Communication Message protocol (DDCMP) is an example of this type of protocol.

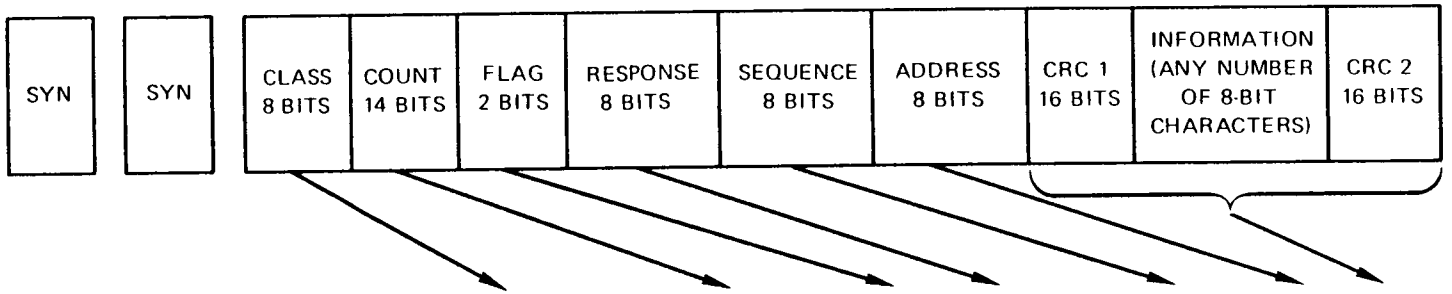
Since data can consist of any bit pattern, it could contain special bit patterns or characters used to frame character or bit oriented protocols. These protocols use special procedures to achieve transparent

data transmitting and receiving. BISYNC uses the technique of inserting another special character whenever it occurs in the data. The receiver checks for duplicate special characters and strips them from the data. DDCMP does away with this time consuming procedure by simply counting the data bytes without checking for special characters. DDCMP was chosen as PREP's data communications protocol because it is very general and easy to use. It provides synchronous or asynchronous, half or full duplex, and serial or parallel communication. It can also support single or multiple transmission stations.

DDCMP uses a very descriptive 8 byte header (Figure 17) to begin its messages. The header's CLASS field indicates three types of messages that provide data (SOH), control (ENQ), and maintenance (DLE) information. Each type of message defines many of the header fields differently. Fields that are commonly defined by each message type are FLAG, ADDRESS, and CRC 1. The FLAG field consists of two bits. One bit is used to inform the receiver that messages are surrounded by sync characters. The other is used in half duplex mode to indicate that the transmitter has sent its last message. The ADDRESS field is the address of each station in multipoint systems. The CRC 1 field is a 2 byte CRC check used to insure proper transmission of the header.

The SOH message is used to indicate data messages. It defines the COUNT field as the number of 8 bit data bytes that follow. The CRC 2 field supplies CRC error checking of the data portion of the message block. The RESPONSE field is the number of the last message received correctly by the sending station. The SEQUENCE field is the number of the current message being transmitted.

ORNL–DWG 81-20892



	XXXXXXXX	XXXXXXXXXXXXXX	XX	XXXXXXXX	XXXXXXXX	XXXXXXXX
SOH – DATA MESSAGES	10000001	CHARACTER COUNT	QS	RESP NO.	MESSAGE NO.	ADDRESS
ENQ ACKNOWLEDGEMENT	00000101	00000001000000	QS	RESP NO.	00000000	ADDRESS
ENQ NEGATIVE ACKNOWLEDGE	00000101	00000010----------	QS	RESP NO.	00000000	ADDRESS

REASONS:	
BCC HEADER ERROR	000001
BCC DATA ERROR	000010
REP RESPONSE	000011
BUFFER UNAVAILABLE	001000
RECEIVER OVERRUN	001001
MESSAGE TOO LONG	010000
HEADER FORMAT ERROR	010001

ENQ REPLY MESSAGE	00000101	00000011000000	QS	00000000	LSTMESS NO.	ADDRESS
ENQ START MESSAGE	00000101	00000110000000	11	00000000	00000000	ADDRESS
ENQ START ACKNOWLEDGEMENT	00000101	00000111000000	11	00000000	00000000	ADDRESS
DLE MAINTENANCE MESSAGE	10010000	CHARACTER COUNT	11	00000000	00000000	ADDRESS

Figure 17. DDCMP Message Format.

ENQ messages consist of the header and are used to control message transmission. The upper 8 bits of the COUNT field are used to determine the type of control. ACKNOWLEDGE (ACK) indicates that messages up to and including the RESPONSE number have been accepted without error. NEGATIVE ACKNOWLEDGE (NAK) implies an erroneous message was received. The last correctly received message number is supplied in the RESPONSE field. The reasons for sending a NAK are contained in the lower 6 bits of the COUNT field. REPLY requests of the receiver the message number of the last correctly received message. The SEQUENCE field is used to supply the last transmitted message number (LSTMESS). The receiver of a REPLY may respond with either a NAK or an ACK. START begins data communications and implies that the next transmitted message number will be 1. START ACKNOWLEDGE (SACK) indicates that a START message has been received.

DDCMP has established procedures for sending and receiving messages. The following are the procedures and rules used by the PREP controller:

1. PREP sends START messages to begin transmission. The receipt of START or a SACK initializes message numbers to 1.
2. Message numbers range cyclicly from 1 to 255.
3. PREP will not accept DLE messages and will respond with a NAK-"Header Format Error".
4. If received, sync characters will be ignored. FLAG and ADDRESS values are also unused.
5. The COUNT value for a SOH message does not include the two data CRC bytes.
6. PREP will reject a message for the following reasons:
   a. Insufficient buffer storage.

b. Message number out of sequence.

c. COUNT values greater than 264.

d. Bad header or data CRC bytes.

7. Any acknowledgement (explicit or implied) with a response number greater than the sequence number of the last message transmitted is ignored.

8. The PREP controller does not send explicit ACKs. When an acknowledgement is desired, it is expected that the host will send a REPLY.

9. The response to a REPLY is always a NAK -"Rep Response". The response value is the last correct message number received.

CRC check bits are the remainder bits left after a modulo-2 polynomial division of an n-byte string. These bits are attached to the end of the string during transmission. The receiver checks for transmission errors by performing the division on the string plus the CRC bits. If the remainder is 0, then no error has occurred.

The calculation of CRC check bits can be performed using either hardware or software. PREP utilizes a software method to generate the 16-bit CRC required by DDCMP. This method is based on the occurrence of a unique remainder for each byte processed through polynomial division. A table of partial remainders is computed and stored for each 8-bit combination. An 8-bit CRC is obtained by looking up the first byte in the table, exclusive oring (EX-OR) its partial remainder with the next byte, and re-entering that value in the table to obtain the next partial remainder. The following bytes are processed by an EX-OR with the previous

partial remainder and finding its result in the table for the next remainder.

To generate a 16-bit CRC value two parallel tables (TH and TL) containing high and low order partial remainders are accessed. Figure 18 shows the algorithm used with these tables. The high order remainder CRC_H is EX-ORed to the next byte while the low order remainder CRC_L is saved for processing with the next byte.

PREP tables TH and TL were generated using the polynomial $X^{16} + X^{15} + X^2 + 1$. These values are stored in ROM on the MSC8004 board.

An interrupt signal initiates PREP's RECEIVE routines (Figure 19). These routines input and process one byte at a time from the host computer. PREP looks for a DLE, ENQ, or SOH byte to initiate communication. The following 7 bytes constitute the header block. Header CRC bytes are checked for correctness. Valid messages are processed either by storing data or sending back a response to the transmitting station. Detection of invalid messages or the completion of correct messages starts the process of looking for a header over again.

Transmitting Data

Transmitting data from PREP to the host computer is similar to receiving data. PREP sends DDCMP data and control messages and follows the same handshaking procedures.

PREP TRANSMIT routines are driven by an interrupt signal that occurs after a character has been transmitted. To initiate the continuous output of characters, PREP loads the first character and enables the 8251 USART for transmission (Figure 20). When no more characters are left, PREP disables transmission so that a garbage character is not transmitted.

ORNL–DWG 81-20893

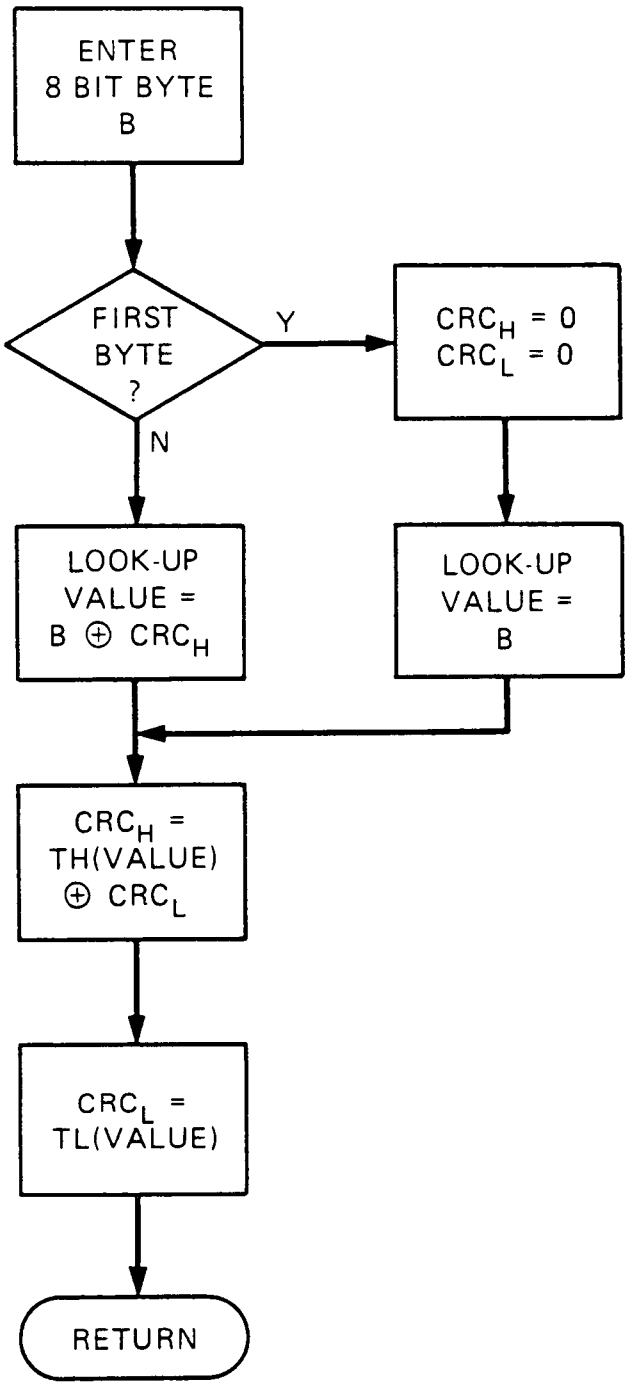


Figure 18. CRC Generation.

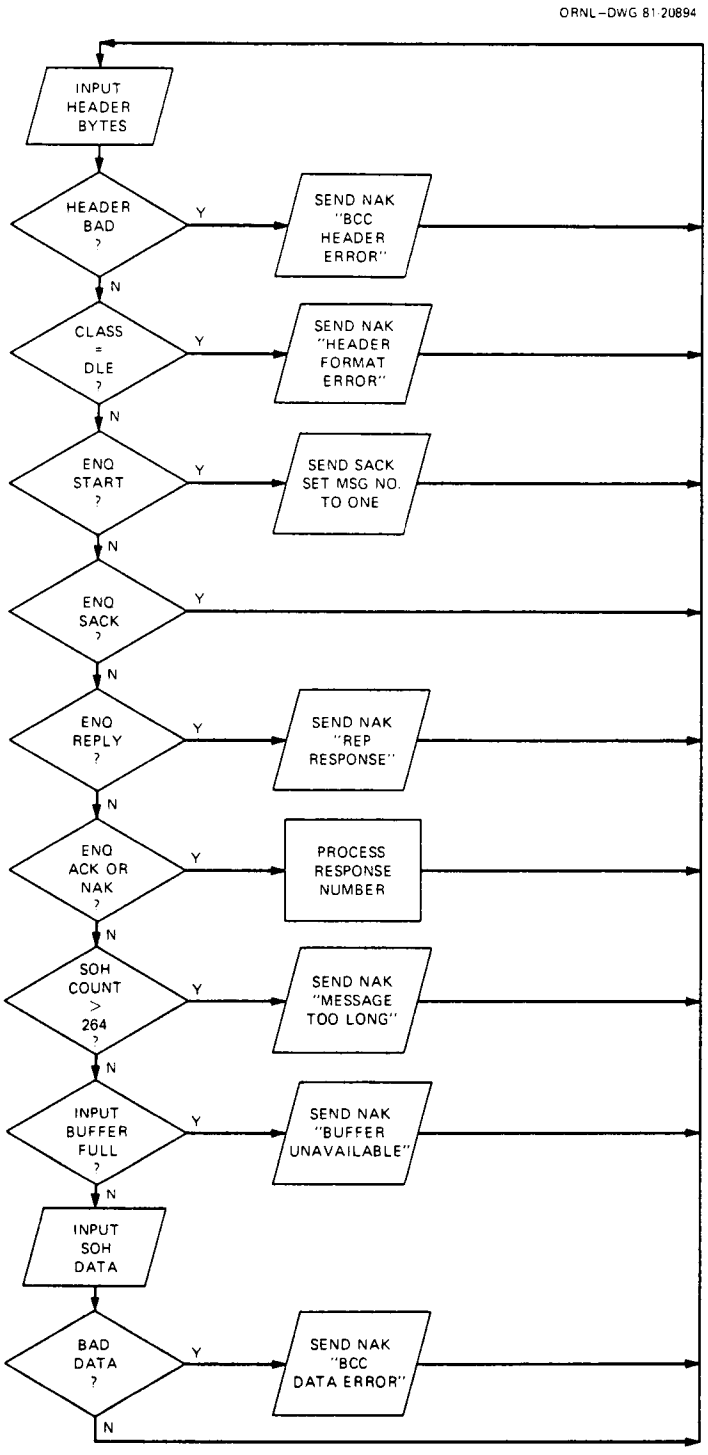


Figure 19. PREP RECEIVE Routine.

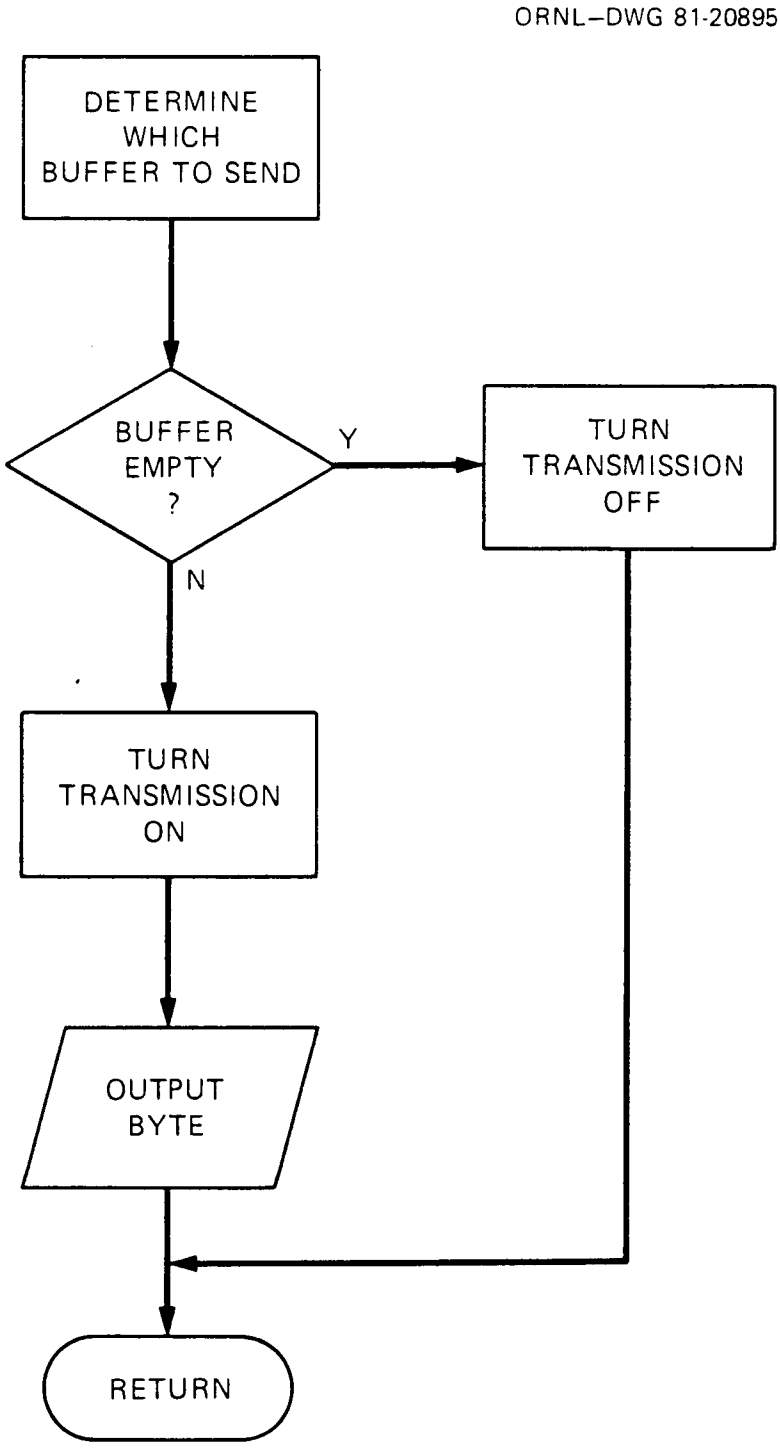


Figure 20. PREP TRANSMIT Routine.

Transmission is activated by loading output buffers with messages. Two ENQ messages are transmitted as the result of the watchdog timer's interrupt signal. After power-up, PREP sets the 8253 timer so that a START message is loaded and sent every second until a START or SACK is received from the host computer. PREP also uses the timer to initiate the transmission of REPLY messages. After a SOH data message is sent to the host computer a REPLY is transmitted every second until an acknowledgement of the data message is received.

I/O Buffer Structure

A complex buffer structure (Figure 21) is used to store input and output messages. Incoming header bytes are accumulated into a sequential input buffer. If the header is determined to be valid by checking the CRC bytes, it is processed according to its CLASS type.

ENQ messages and bad CRC header or data calculations require PREP to respond to the host computer with the appropriate ENQ response. These messages are formed by the RECEIVE routine and stored into Output Buffer 0.

SOH data is stored into the Input First-In-First-Out (FIFO) Buffer. The FIFO buffer is controlled by four pointers. These pointers are the addresses of the beginning and the end of the buffer, the oldest byte in the buffer, and the next free space. Two status parameters are also used to determine whether the buffer is full or empty. As shown in Figure 22, putting a byte into the FIFO buffer is accomplished by manipulating the pointers. Before filling the buffer with a new data string, these pointers are saved in temporary storage. After checking CRC data bytes, bad data are easily removed from the buffer by restoring the pointers to the position before the data was added.

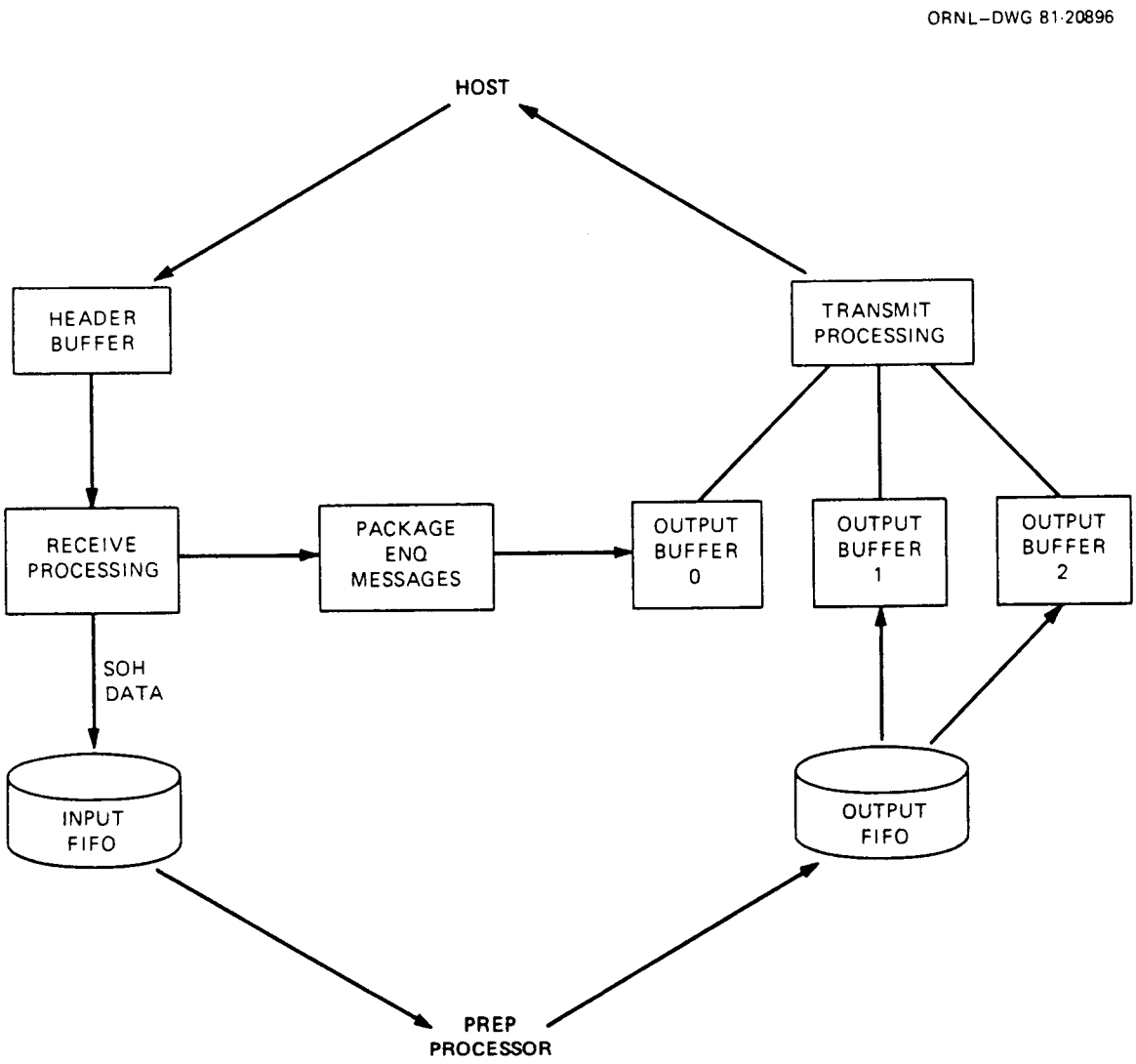


Figure 21. PREP Buffer Structure.

ORNL–DWG 81-20897

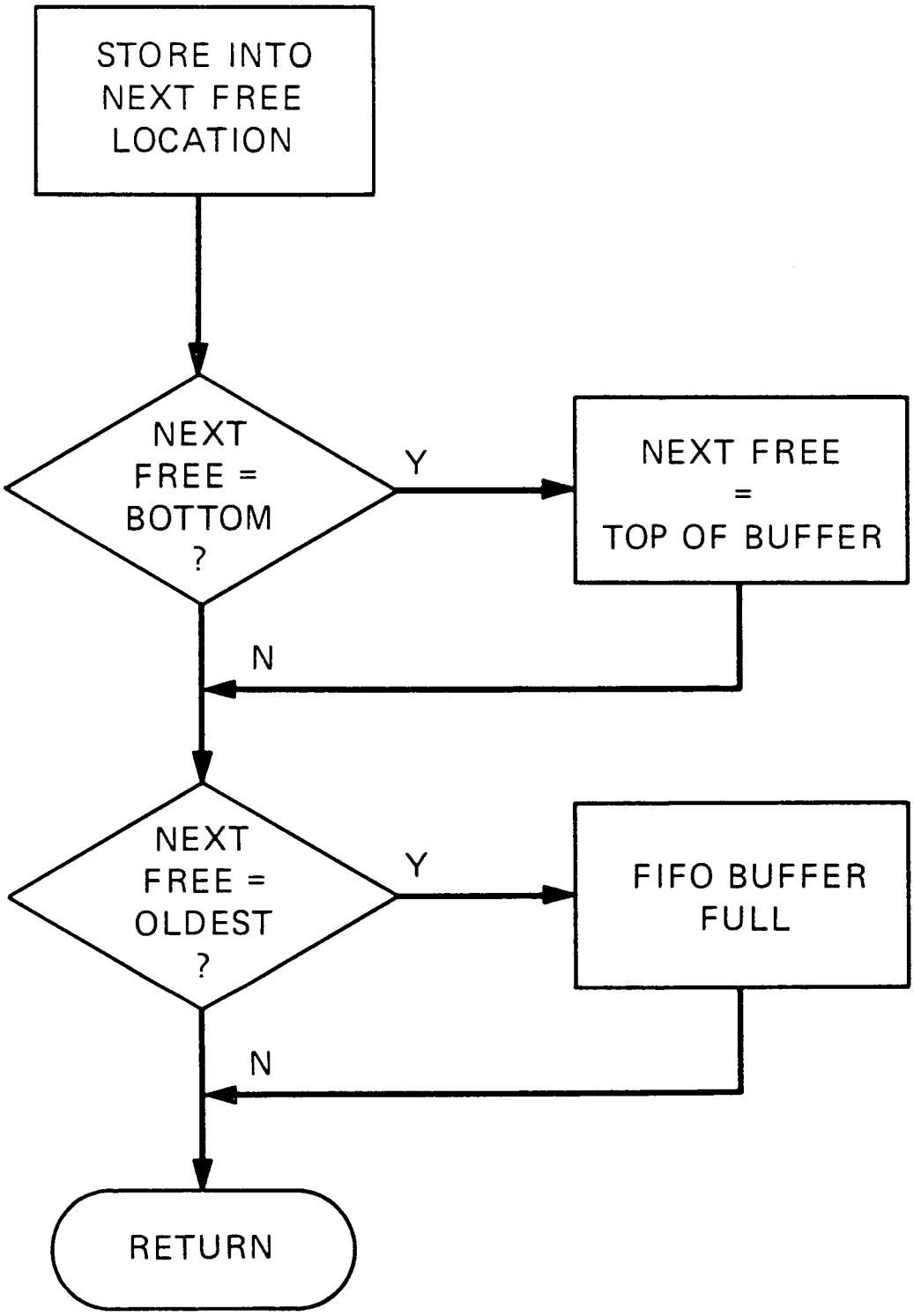


Figure 22. FIFO Buffer Storage.

During periods when PREP is not under interrupt control, the PREP processor is used to interpret the data and perform specified actions. Most of the actions involve the plotter and the slave microprocessor and will be discussed in detail in the following chapter. Some actions cause SOH data messages to be sent back to the host computer. These messages are prefaced with a DDCMP header and stored into the Output FIFO buffer. Storing data into a FIFO buffer is accomplished in much the same way as retrieving data. SOH messages are transferred from the FIFO buffer into one of two sequential output buffers. After a message has been transmitted from either Output Buffer 1 or 2, the contents of the buffers are retained and retransmitted until an acknowledgement from the host is received.

The TRANSMIT routine chooses which of the three sequential buffers to send. Buffer 0 contains the most pressing information (ENQS) and is given the highest priority. Two flags are used to determine whether Buffer 1 or 2 is sent next. One flag is used to control when the buffer contents should be transmitted or retransmitted so that the host does not receive duplicate messages. The other flag indicates the buffer's message number. It is checked before transmission so that message N+1 is not transmitted before message N. Figure 23 depicts how the TRANSMIT routine uses the flags to determine which buffer to send.

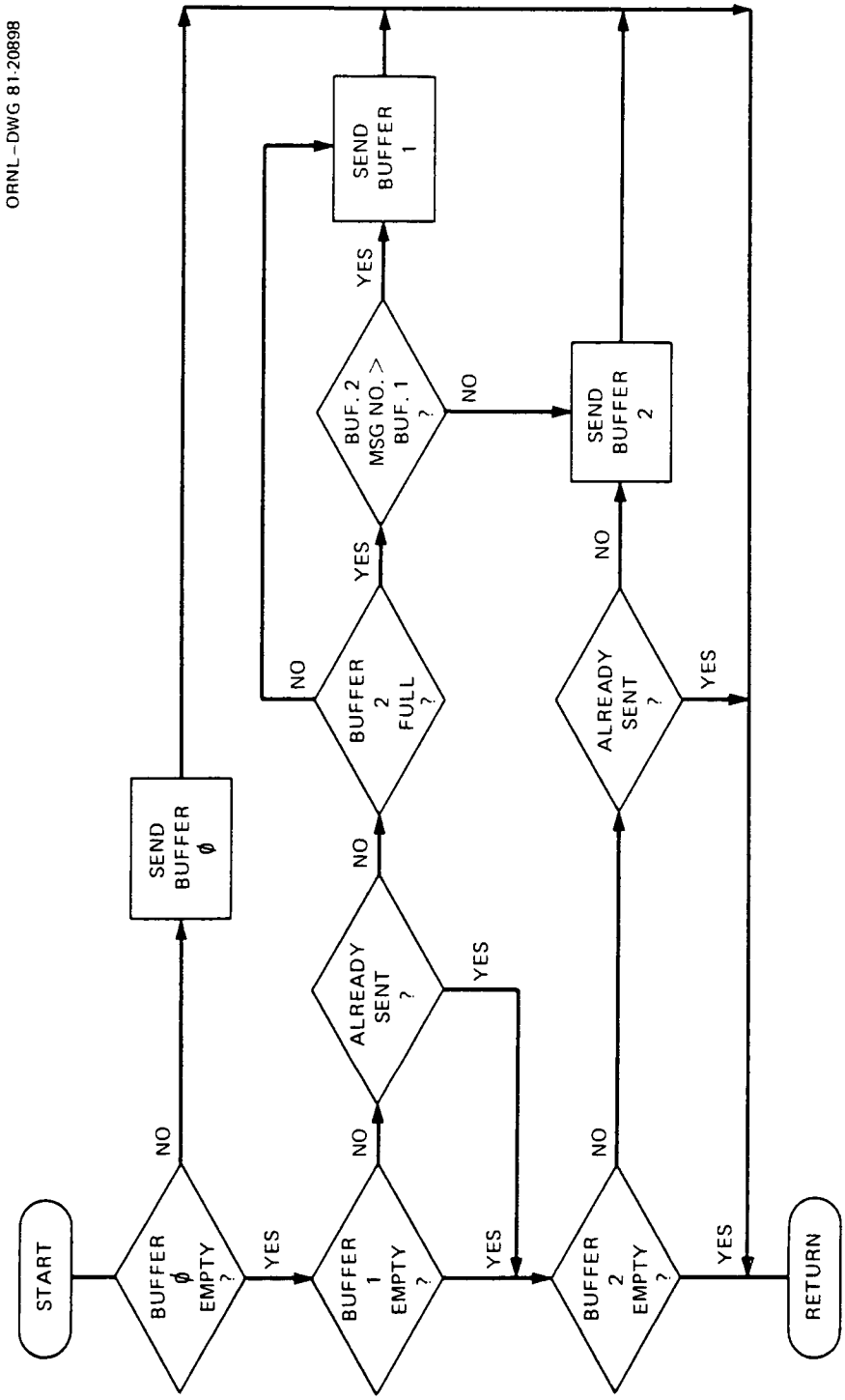


Figure 23. Determination of Transmitted Buffer.

CHAPTER V

THE PREP PROCESSOR

The PREP PROCESSOR controls all background tasks not initiated by interrupts. The processor continually retrieves data messages from the Input FIFO Buffer and interprets what action is to be performed. Data messages (Figure 24) from the host consist of a Command Id byte followed by 0 to 264 bytes of information. This information can be print data, plot data, compressed plot data, or executable programs. As shown in Table 2, commands from the host computer cause actions to occur with respect to the plotter, the MSC8004, and the Z8000. Appendix D defines PREP command formats.

Plotter Commands

The host sends a STATUS Command Id to ask PREP for the current plotter status. Upon receiving this command, the PREP processor loads the Output FIFO buffer with a STATUS Command Id byte followed by the plotter status byte. The status byte is determined after PREP reads the VERSATEC status lines ONLIN and NOPAP. When the VERSATEC is on and not out of paper the status byte is zero. A non-zero status byte indicates a problem with the plotter. If a problem occurs while printing or plotting, the PREP PROCESSOR will send the status message to the host without receiving a STATUS command.

The FORM FEED command causes the VERSATEC to move the paper to the next top of page. This command is used to finish the current page or output a blank page. A FORM FEED command issued before each print/plot job produces a job separator page. It also ensures that the first page

ORNL–DWG 81-20899

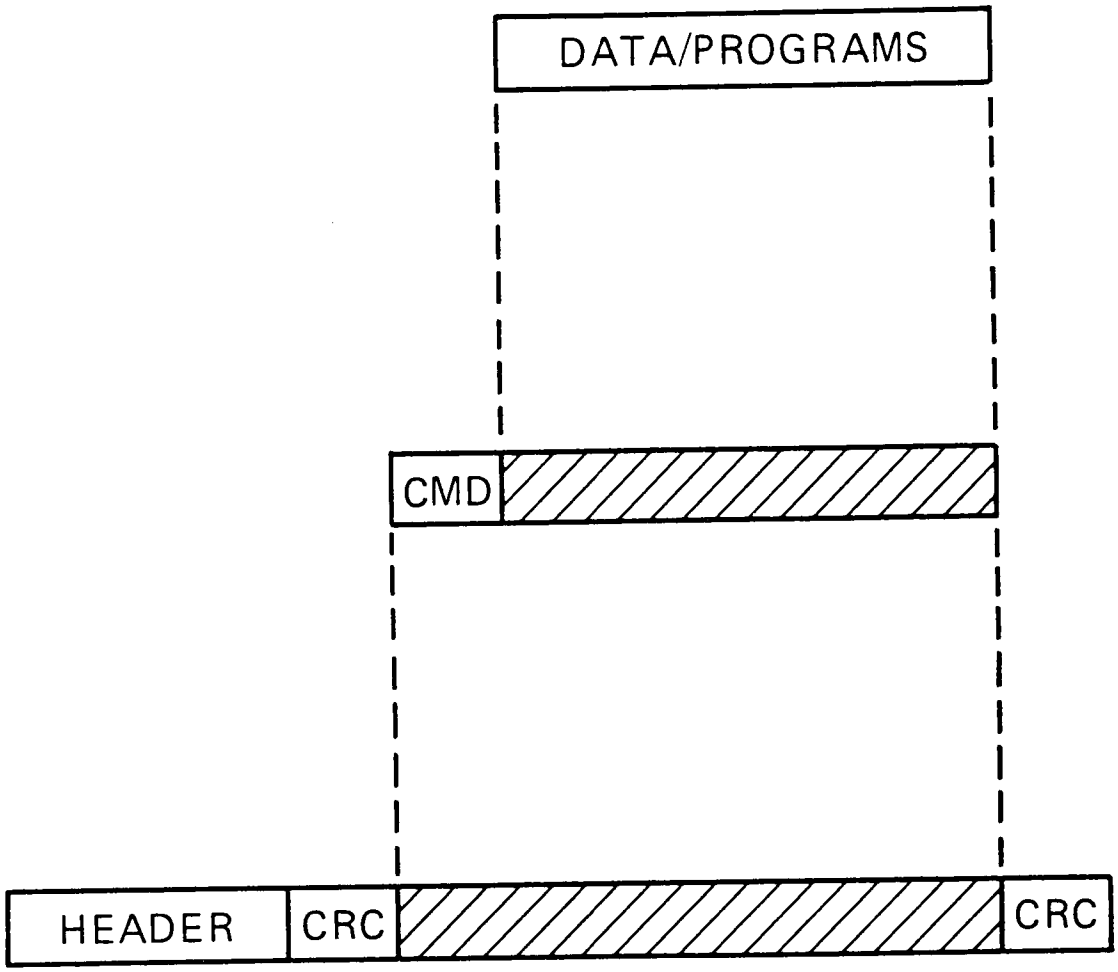


Figure 24. PREP Data Message.

TABLE 2

PREP Commands

Command	Action
STATUS	Requests plotter status
FORM FEED	Moves paper to top at page
PRINT	Prints ASCII characters
PLOT IMMEDIATE	Plots uncompressed data
PLOT COMPRESSED	Plots compressed data
STORE Z80	Stores binary data in MSC8004
STORE Z8000	Stores binary data in Z8000
EXECUTE Z80	Executes at MSC8004 location
EXECUTE Z8000	Executes at Z8000 location
ABORT Z80	Aborts MSC8004 processing
ABORT Z8000	Aborts Z8000 processing

printed or plotted is clean. A first page saturated with toner can occur after the VERSATEC is inactive.

The PRINT Command Id is followed by ASCII characters. A full print line consists of 132 characters. Partial lines may be terminated by sending the ASCII codes listed in Table 3. LF and VT cause a single line feed. EOT and FF advance the paper to the top of the page. Figure 25 shows how a sample ASCII print line is interpreted.

TABLE 3

PREP Print Line Terminators

ASCII Code	HEX Value
LF (Line Feed)	0A
VT (Vertical Tab)	0B
FF (Form Feed)	0C
EOT (End of Transmission)	04

```
[FF] A QUICK BROWN[CRLF] FOX[VTVT] JUMPS OVER[LF] THE[LF] LAZY[EOT] DOG.[LF]

                 ┌──────────────────────────────────────────────────────┐
TOP OF PAGE      | A QUICK BROWN                                        |
LINE 2           | FOX                                                  |
     3           |                                                      |
     4           | JUMPS OVER                                           |
     5           | THE                                                  |
     6           | LAZY                                                 |
     .           |                                                      |
     .           |                                                      |
     .           |                                                      |
END OF PAGE      |                                                      |
                 └──────────────────────────────────────────────────────┘

                 ┌──────────────────────────────────────────────────────┐
TOP OF PAGE      | DOG.                                                 |
                 |                                                      |
                 |                                                      |
                 |                                                      |
                 |                                                      |
                 |                                                      |
                 |                                                      |
                 |                                                      |
END OF PAGE      |                                                      |
                 └──────────────────────────────────────────────────────┘
```

Figure 25. PREP Processing of ASCII Print Data.

PREP places the VERSATEC in print mode using the control PRINT line. Print lines are processed one at a time. ASCII characters fill a buffer until a line terminator character is encountered, or until 132 characters have arrived. The READY status line is checked before each byte in the buffer is sent to the VERSATEC. The remote line feed control RLTER is used to terminate the line when the buffer is empty.

The PLOT Command Id precedes bytes of uncompressed data. A full plot line consists of 264 bytes. Since there are no line terminators for uncompressed plot data, only one line is generated per command. Plot data bytes are stored in a buffer until the end of the command string is reached. The buffer is sent byte by byte to the VERSATEC in the same manner as print data.

The PLOT COMPRESSED command is used to transfer compressed plot data. The bytes that follow the ID byte are formulated according to a scheme developed by Lawrence Livermore Laboratory. This scheme segments plot data into 0 to 64 byte groups. A control byte precedes each group. The two high order bits of this byte specify how the data was compressed. The low order six bits usually contain a count. There are six kinds of control bytes in this compression technique. Formats for each type of byte are given in Appendix E. The control bytes describe data that has been compressed horizontally and vertically. Horizontal compression is obtained by eliminating repeating horizontal bytes. Repeating bytes of zeroes are replaced by a control byte containing the number of repeats. The control byte for non-zero repeating bytes is followed by the repeated byte. The control byte contains the repeat count. Vertical compression eliminates repeating lines and repeating vertical bits. A control byte is used to

describe the number of times to repeat the last plot line. Vertical bits are eliminated by EX-ORing the last line with the current line. The resulting line and the current line are then compressed. If the resulting line is shorter than the current line then it is sent. The control byte is used to flag the line as the result of the EX-OR technique. Page and line formatting are also controlled by control bytes. These bytes are used to specify line and page termination.

Z80/Z8000 Commands

The STORE Z80/Z8000 commands insert data or executable programs into the MSC8004/Z80 or COMPUTEX Z8000 memory respectively. The Command Id byte is followed by the first location to store the data. This address field may either contain a valid address or 0000H. If an invalid address occurs, the data will not be stored. MSC8004 addresses range from 8000H to DFFFH. Valid addresses for the Z8000 are 0001H to 7FFFH. A STORE address of 0000H is used to continue storing data at consecutive locations.

The EXECUTE Z80/Z8000 commands are used to transfer execution to a given location. An optional limit of up to 65 seconds can be placed on execution time to prevent lockout. PREP uses the watchdog timer to generate an interrupt after the specified time. Control is transferred back to the processor. Normal program control should be transferred back via a Z80/Z8000 RETURN statement.

Execution can also be halted with the ABORT Z80/Z8000 command. The ABORT Z8000 command halts all processing and executing by the Z8000. The ABORT Z80 commands halts all command processing and plotting controlled by the MSC8004. It also disables the receiving and transmitting of data while the PREP PROCESSOR empties the Input/Output FIFO buffers. After completion

of an ABORT Z80, the PREP controller is returned to a state where it is ready to accept the next DDCMP message from the host computer.

CHAPTER VI

FUTURE PLANS

Since the PREP controller is a flexible and usable tool, it will be used for many applications. PREP controllers will be installed at several Oak Ridge National Laboratory sites. These controllers will be interfaced to the ORNL PDP 11/70s and be used primarily as remote word processing output devices. Presently, word processing programs in the 11/70 computers have been modified to send data to remote locations. One PREP controller has been tested quite thoroughly and appears to be very reliable as a remote print/plot device. The process of transforming computer code that can be executed on the Z8000 microcomputer is also in progress.

Another installation in Oak Ridge has plans to interface PREP with a VAX computer. This facility will also use PREP for its word processing capabilities.

Besides plotting textual information, the PREP controller will be used for graphs. A group of programmers at ORNL plan to send compressed plot files to PREP. These files are created on a PDP-10 using a large interactive graphics package. A computer program will be developed to decompress the files. This program will be downloaded and executed using the PREP processor commands.

Programmers at the Oak Ridge Y-12 plant have opted to link the PREP controller with a PDP 11/45 computer. The 11/45 will receive compressed plot data (graphs, charts, etc.) from a CRAY computer located at the Lawrence Livermore Laboratory in California. Very few additions are needed to the PDP 11/45 to transmit data to PREP. The plot data is compressed

using the same scheme employed by PREP and is sent to the computer using DDCMP protocol.

Since PREP will be used heavily by many installations, thought has been given to improving the controller. Future improvements could involve both hardware and software modifications. Adding direct memory access capabilities to the MSC8004 would allow the plotter to be driven at full speed. Processing incoming data from the host computer could be optimized by adding hardware FIFO buffers and CRC generators. Compression of print data would speed print job transmission time. Diagnostic routines for testing serial and parallel I/O channels would be helpful in tracing hardware failures.

Most of these modifications would add extra expense to the cost of the controller but speed up PREP processing times. Presently, the extra speed is not necessary when interfaced to the ORNL 11/70 computer. Since the computer is operating at full capacity, the PREP controller can process data faster than it is transmitted. These improvements should prove to be valuable when the PREP controller is linked with other host computers and print/plot devices.

LIST OF REFERENCES

LIST OF REFERENCES

1. B.A. Artwick, Microcomputer Interfacing, Prentice-Hall,Inc., 1980.

2. J.E. McNamara, Technical Aspects of Data Communication, Digital Press, 1977.

3. R.L. Evans, "Communication Protocol - The Key to Successful Data Communications", Computer Technology Review, Spring/Summer, 1981.

4. Essentials of Data Communication, TEKTRONIX Inc., 1978.

5. B. Soucek, Microprocessors and Microcomputers, John Wiley and Sons, 1976.

6. PDP-11 Processor Handbook, Digital Equipment Corporation, 1979.

7. INTERACTIVE System/One Programmer's Manual, INTERACTIVE Systems Corporation, February, 1981.

8. User's Manual for MSC8004, Monolithic Systems Corporation, 1978.

9. L. Smith, Using the 8251 Universal Synchronous/Asynchronous Receiver/Transmitter, INTEL Application Note AP-16, INTEL Corporation, 1976.

10. A. Ebright, 8255A Programmable Peripheral Interface Applications, INTEL Application Note AP-15, INTEL Corporation, 1976.

11. L. DeJong, et al., "Microcomputer Interfacing Characteristics of the 8253 Programmable Interval Timer", Computer Design, February, 1978.

12. Z80 Assembly Language Programming Manual, ZILOG, Inc., 1977.

13. Z80 CPU Technical Manual, ZILOG, Inc., 1977.

14. INTEL MULTIBUS Specification, Manual No. 9800683, INTEL Corporation, 1978.

15. ZILOG Z8000, An Architectural Overview, ZILOG, Inc., 1978.

16. Hardware Reference Manual for the Z8000 CPU Board, COMPUTEX Company, 1981.

17. Hardware Reference Manual for the 32K-128K Dynamic RAM Board, COMPUTEX Company, 1981.

18. Hardware Reference Manual for the MULTIBUS Mother Board, COMPUTEX Company, 1981.

19. VERSATEC V-80 Series Product Description Manual, Publication No. 5939-01, May, 1980.

20. An Introduction to Microcomputers, Volume 3, Osborne and Associates Inc., 1979.

21. R.H. Frobose, Jr., Compression of Graphic Data in Raster Format, UCRL-51858, Lawrence Livermore Laboratory, 1975.

22. S. Wecker, A Table Algorithm for Software Computation of Cyclic Redundancy Check, Digital Equipment Corporation, 1974.

23. J.S. Whiting, "An Efficient Software Method for Implementing Polynomial Error Detection Codes", Computer Design, March, 1975.

24. 8002A Microprocessor Lab System User's Manual, TEKTRONIX, 1978.

25. 8002A Microprocessor Lab Editor Manual, TEKTRONIX, 1980.

APPENDICES

APPENDIX A

PREP SERIAL I/O CONFIGURATION

TABLE A-1

Serial I/O Installation

Board Location	Part	Function
U19	8251	USART
U17	8253	Programmable Timer
U9	1488	Line Driver
U10	1489	Line Receiver

TABLE A-2

Serial I/O Jumpers

Pin Connections	Function
40-42	Internal timer clock
43-44	-12 v to driver/receiver
46-47	+12 v to driver/receiver
50-51	2 MHz clock to 8253 and 8251
96-97	Supplies DSR
98-99	Txc = Rxc

TABLE A-3

Serial I/0 Cable Connections*

MSC8004 J3 Edge Connector Pin #	Signal	Female RS-232 Cable Pin #	Signal
5	TxD	3	RxD
3	RxD	2	TxD
13	GND	7	GND
11	DTR	6	DSR
9	RTS	5	CTS
7	CTS	4	RTS
14	DSR	20	DTR

* In some applications, DSR, CTS, RTS, and DTR are not supplied by the host computer. For these cases, tie the MSC8004 DTR to DSR and RTS to CTS.

TABLE A-4

Serial I/0 Address Control

Part	Address	Read	Write
8251	ED	Status	Control/Mode
8251	EC	Data In	Data Out
8253	DC	Timer 0 Status	Timer 0 Count
8253	DD	Timer 1 Status	Timer 1 Count
8253	DE	Timer 2 Status	Timer 2 Count
8253	DF	Invalid	Control

APPENDIX B

PREP PARALLEL I/O CONFIGURATION

TABLE B-1

Parallel I/O Installation

Edge Connector	Board Location	Part	Function
J1	U16	8255	Peripheral Interface
J1	U4	8287	Inverting Transceiver
J1	U3	74368	Inverting Buffer
J1	U2	8286	Non-inverting Transceiver
J1	RN2-RN4	7924	4.7k Resistors
J2	U18	8255	Peripheral Interface
J2	U8	8303	Inverting Transceiver
J2	U7	7937	Inverting Buffer
J2	U6	8303	Inverting Transceiver
J2	RN5-RN7	8110	4.7k Resistors

TABLE B-2

Parallel I/O Address Control

	Address		
Port	J1	J2	Function
A	E4	E8	Read/Write
B	E5	E9	Read/Write
C	E6	EA	Read/Write
Control	E7	EB	Status/Control

TABLE B-3

Parallel I/O Port Configuration

Edge Connector	Port	Bit #	VERSATEC Signal
J1	A	0	CLEAR
J1	A	1	RESET
J1	A	2	RFFED
J1	A	3	REOTR
J1	A	4	RLTER
J1	B	0	IN01
J1	B	1	IN02
J1	B	2	IN03
J1	B	3	IN04
J1	B	4	IN05
J1	B	5	IN06
J1	B	6	IN07
J1	B	7	IN08
J1	C	1	PICLK
J2	A	0	PRINT
J2	A	1	SPP
J2	A	2	PARIN
J2	B	0	READY
J2	B	1	NOPAP
J2	B	2	ONLIN

TABLE B-4

Parallel I/0 Cable Connections

MSC8004			VERSATEC		
Edge Connector	Signal Pin #	Ground Pin #	Signal Pin #	Ground Pin #	Signal
J1	48	47	9	28	CLEAR
J1	46	45	15	34	RESET
J1	44	43	16	35	RFFED
J1	42	41	17	36	REOTR
J1	40	39	18	37	RLTER
J1	16	15	1	20	IN01
J1	14	13	2	21	IN02
J1	12	11	3	22	IN03
J1	10	9	4	23	IN04
J1	8	7	5	24	IN05
J1	6	5	6	25	IN06
J1	4	3	7	26	IN07
J1	2	1	8	27	IN08
J1	22	21	10	29	PICLK
J2	48	47	12	31*	PRINT
J2	46	45	14	33	SPP
J2	44	43	13	31*	PARIN
J2	16	15	11	30	READY
J2	14	13	19	**	NOPAP
J2	12	11	32	**	ONLIN

*Two grounds tied to Pin 31.
**No ground for these signals.

TABLE B-5

Parallel I/0 Jumpers

Pin Connections	Function
20-22	ACK = 0BF
29-30	Port B used as input

APPENDIX C

PREP JUMPER CONFIGURATION

TABLE C-1

Interrupt Jumpers

Pin Connections	Functions
3-80	TxRDY to 8214
4-82	RxRDY to 8214
1-78	Timer No. 1 to 8214
2-27	Out0 to CLK1
29-26	GND to GATE1
24-28	2 MHz CLK to CLK0
26-25	GATE0 to GATE1

TABLE C-2

Multibus Jumpers

Pin Connections	Function
60-61	Reset generator to INIT
62-63	8MHz to BCLK
68-69	8MHz to CCLK
117-118	GND to BPRN

TABLE C-3

Other Jumpers

Pin Connections	Function
58-59	4MHz to 280 processor
90-91	Enables 16x1 RAM
94-95	Enables 16x1 RAM
109-110	+12v to RAM
111-112	+5v to RAM
114-115	-5v to RAM
100-101	Reduces wait state

APPENDIX D

PREP COMMAND FORMATS

Command: STATUS

Format:



Command Id: 00H

Action: Asks PREP for the current plotter status

Response Format:



Command Id: 00H

Plotter Status Byte: Bits 0-5 = Not Used. Set to 0.
Bit 6 = NOPAP
Bit 7 = ONLIN

Response Action: PREP answers by sending the same Command Id followed by a status byte. When the plotter is functioning properly, the status byte is 0. Bit 7 is set when the plotter is not online. Bit 6 is set if the plotter is out of either paper or toner.

Command: FORM FEED

Format:

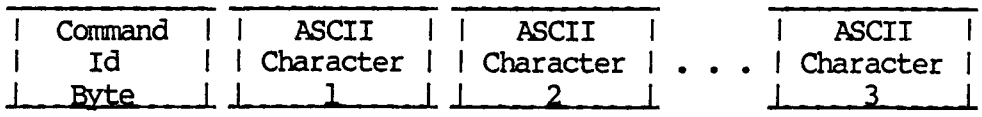


Command Id: 01H

Action: Causes the paper to move to the top of the page.

Command: PRINT

Format:

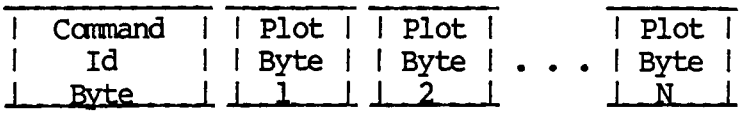


Command Id: 02H

Action: PREP prints N ASCII characters. N is determined by protocol definition.

Command: PLOT IMMEDIATE

Format:

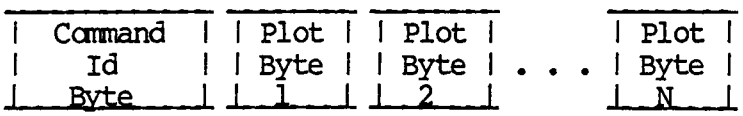


Command Id: 03H

Action: PREP plots N bytes of raster plot data. The number of plot bytes per command is determined by protocol.

Command: PLOT COMPRESSED

Format:



Command Id: 04H

Action: PREP decompresses the N compressed plot bytes. The number of plot bytes per command is determined by protocol.

Command: STORE Z80

Format:

Command Id Byte	High Order Address Byte	Low Order Address Byte	Plot Byte 1	Plot Byte 2	. . .	Plot Byte N

Command Id: 05H

Address: 8000H - DFFFH, or 0000H.

Action: Store N bytes of data into the MSC8004 starting at the address location. If 0000H is specified, the next consecutive address is used.

Command: STORE Z8000

Format:

Command Id Byte	High Order Address Byte	Low Order Address Byte	Plot Byte 1	Plot Byte 2	. . .	Plot Byte N

Command Id: 06H

Address: 0000H - 7FFFH

Action: Stores N bytes of data into the COMPUTEX Z8000 starting at the address location. If 0000H is specified, the next consecutive address is used.

Command: EXECUTE Z80

Format:

Command Id Byte	High Order Address Byte	Low Order Address Byte	High Order Time Byte	Low Order Time Byte

Command Id: 07H

Address: 8000H - DFFFH

Time: 0000H - FFFFH

Action: Executes code in the MSC8004 starting at the address location. An optional time limit specified in milliseconds can be used to abort program execution. If the time limit is zero, execution will not be terminated.

Command: EXECUTE Z8000

Format:

Command Id Byte	High Order Address Byte	Low Order Address Byte	High Order Time Byte	Low Order Time Byte

Command Id: 08H

Address: 0001H - 7FFFH

Time: 0000H - FFFFH

Action: Executes code in the Z8000 starting at the address location. An optional time limit specified in milliseconds can be used to abort execution. If the time limit is zero, execution will not be terminated.

Command: ABORT Z80

Format:



Command Id: 09H

Action: Aborts any processing controlled by the MSC8004. This includes current command processing, execution of user programs, and print/plot operations. Protocol message numbering is unaffected.

Command: ABORT Z8000

Format:



Command Id: 0AH

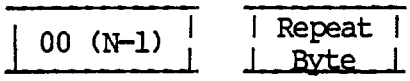
Action: Aborts any processing controlled by the Z8000.

APPENDIX E

PREP COMPRESSION FORMATS

Control Byte: 000B - 077B

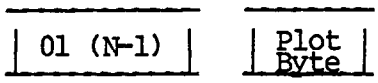
Format:



Action: REPEATS the byte following the control byte N times. N ranges from 2 to 64. A zero control byte indicates a form feed.

Control Byte: 100B - 177B

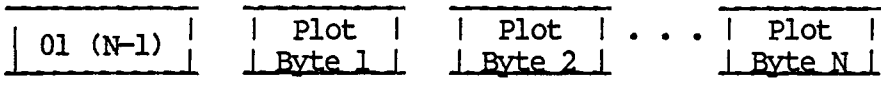
Format:



Action: Skip N bytes and plot the byte following the control byte. N ranges from 1 to 64.

Control Byte: 200B - 277B

Format:



Action: Plot the next N bytes following the control byte. N ranges from 1 to 64.

Control Byte: 300B

Format:

| 11000000 |

Action: Terminates a plot line.

Control Byte: 301B - 336B

Format:

| 11 (N) |

Action: Repeat the previous line N more times. N ranges from 1 to 30.

Control Byte: 337B

Format:

| 11011111 |

Action: The present line is terminated and the following line is to be EX-ORed with the previous line.

VITA

Janet H. Carney was born in Oak Ridge, Tennessee on September 10, 1952. She attended public schools in Clinton, Tennessee and graduated from Clinton High School in June, 1970.

The fall of 1970 she entered The University of Tennessee in Knoxville. After four years she was awarded a Bachelor of Science degree with a major in Mathematics.

In 1974, Ms. Carney was employed by the Oak Ridge National Laboratory as a Computer Analyst. While at the laboratory, she has worked on many energy and environmental projects and co-authored many articles in these fields. Currently, Ms. Carney is a systems programmer for the ORNL Text Management group.

In addition to her work at ORNL, Ms. Carney began taking Computer Science graduate courses at The University of Tennessee. She graduated with a Computer Science Master's degree in the fall of 1981.

Ms. Carney is a member of Phi Beta Kappa and the Association for Computing Machinery.